Rossana Henriques dos Santos

Mestre em Engenharia Informática

Augmenting Spaces and Creating Interactive Experiences Using Video Camera Networks

Dissertação para obtenção do Grau de
Doutor em Engenharia Informática

Orientador: Professor Doutor Nuno Manuel Robalo Correia,
Prof. Catedrático, Universidade Nova de Lisboa

Júri:

Presidente:	Professor Doutor José Legatheaux Martins
Arguentes:	Professora Doutora Mari Carmen Juan Lizandra
	Professor Doutor Adérito Fernandes Marcos
Vogais:	Professor Doutor João António Madeiras Pereira
	Professor Doutor Nuno Manuel Ribeiro Preguiça
	Professor Doutor Nuno Manuel Robalo Correia

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Setembro, 2015

Augmenting Spaces and Creating Interactive Experiences Using Video Camera Networks

Copyright © Rossana Henriques dos Santos, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

To my parents, Miguel, Hugo and Matilda.

Acknowledgements

I first wish to thank my supervisor, Prof. Nuno Correia, for his advice, dedication and support through out this years as well as for arranging all the leisure moments that make our group such a nice place to work at.

I would like to thank all my colleagues and professors at the NOVA Laboratory for Computer Science and Informatics, especially my present and former colleagues at the Interactive Multimedia Group (IMG), André Sabino, Bruno Cardoso, Diogo Cabral, Duarte Gonçalves, Filipa Peleja, Pedro Centieiro, Rui Madeira, Rui Nóbrega, Rui Jesus, Sofia Reis, Filipa Peleja, Tarquinio Mota and Tiago Martins for their help, suggestions and support. To Prof. Teresa Romão, Prof. Eduardo Dias, Prof. Rui Prada and Prof. Mónica Mendes for valuable input and feedback. To all the staff at FCT. To all my colleagues (a special thank you to Marta Ferraz) and teachers at the UT Austin | Portugal Program and all the friends I made during my stay at Austin, especially Barbara Gallardo.
I also would like to thank to Prof. Carmen Morgado for the dancing, lunches, breaks and awesome cakes and cookies.

All these years of research could not be possible without the support, motivation and patience of my parents and family. A big thank you to Miguel and my son Hugo, who was born on the final stretch of this work and to all my four legged friends for the so needed recreational moments.

I would like to express my thanks to all my colleagues and friends at the Escola Superior de Tecnologia de Setúbal and at the Erasmus Intensive Programme SDI who stood by my side and gave me the motivation to keep going. A special thank you to Prof. Joaquim Filipe who's help and collaboration was essential to give me the room and peace of mind necessary to complete this work.

This work is partly funded by the Fundação para a Ciência e a Tecnologia, Ministério da Educação e da Ciência, Portugal, grant SFRH/BD/46980/2008 of the UT Austin | Portugal Program and by the NOVA Laboratory for Computer Science and Informatics (NOVA LINCS), Multimodal Systems, Departamento de Informática (DI), Faculdade de Ciências e Tecnologia (FCT), Universidade Nova de Lisboa (UNL) and Escola Superior de Tecnologia de Setúbal (EST Setúbal), Instituto Politécnico de Setúbal (IPS).

ABSTRACT

This research addresses the problem of creating interactive experiences to encourage people to explore spaces. Besides the obvious spaces to visit, such as museums or art galleries, spaces that people visit can be, for example, a supermarket or a restaurant.

As technology evolves, people become more demanding in the way they use it and expect better forms of interaction with the space that surrounds them. Interaction with the space allows information to be transmitted to the visitors in a friendly way, leading visitors to explore it and gain knowledge.

Systems to provide better experiences while exploring spaces demand hardware and software that is not in the reach of every space owner either because of the cost or inconvenience of the installation, that can damage artefacts or the space environment.

We propose a system adaptable to the spaces, that uses a video camera network and a wi-fi network present at the space (or that can be installed) to provide means to support interactive experiences using the visitor's mobile device.

The system is composed of an infrastructure (called vuSpot), a language grammar used to describe interactions at a space (called XploreDescription), a visual tool used to design interactive experiences (called XploreBuilder) and a tool used to create interactive experiences (called urSpace).

By using XploreBuilder, a tool built of top of vuSpot, a user with little or no experience in programming can define a space and design interactive experiences. This tool generates a description of the space and of the interactions at that space (that complies with the XploreDescription grammar). These descriptions can be given to urSpace, another tool built of top of vuSpot, that creates the interactive experience application.

With this system we explore new forms of interaction and use mobile devices and pico projectors to deliver additional information to the users leading to the creation of interactive experiences. The several components are presented as well as the results of the respective user tests, which were positive.

The design and implementation becomes cheaper, faster, more flexible and, since it does not depend on the knowledge of a programming language, accessible for the general public.

Keywords: Development Tools; Mobile Device Applications; Augmented Reality; Space Exploration; Interaction; Gaming

RESUMO

Nesta dissertação é abordado o problema do desenvolvimento e implementação de aplicações usadas para incentivar pessoas a explorar espaços. Para além dos espaços onde tradicionalmente se realizam visitas, tais como museus e galerias de arte, outros espaços podem ser considerados, tais como, supermercados ou restaurantes.

Com a evolução da tecnologia, as pessoas tornaram-se mais exigentes na forma de usá-la e esperam melhores formas de interação com o espaço que as rodeia. A interação com o espaço permite que informação adicional possa ser transmitida aos seus visitantes, levando-os a explorá-lo e a enriquecer o seu conhecimento.

Sistemas para oferecer melhores experiências enquanto se explora um espaço exigem hardware e software que não está ao alcance de todos os proprietários ou gestores de espaços, pelo custo associado ou pela inconveniência de instalação de hardware e outros equipamentos, que pode danificar artefactos ou património.

Neste documento propomos um sistema que utiliza redes de câmaras de vídeo e redes de comunicação sem fios para suportar experiências interativas usando os dispositivos móveis dos visitantes. O sistema é composto por uma infraestrutura (chamada vuSpot), uma gramática (chamada XploreDescription), uma ferramenta visual para desenhar experiências interativas (chamada XploreBuilder) e uma ferramenta que cria a aplicação correspondente à experiência interativa descrita (chamada urSpace).

Utilizando a ferramenta visual XploreBuilder, um utilizador com pouca ou nenhuma experiência de programação pode definir um espaço e desenhar experiências interativas. Esta ferramenta gera a descrição do espaço e a descrição das interações da experiência interativa, que obedece às regras da XploreDescription. Estes ficheiros podem ser passados à ferramenta urSpace, que foi desenvolvida utilizado a infraestrutura vuSpot e que cria a aplicação correspondente à experiência interativa descrita.

Neste sistema utilizamos dispositivos móveis e pico projectores na criação de experiências interactivas e pretendemos explorar novas formas de interacção. Os componentes do sistema são descritos e os respectivos resultados dos testes de utilizador, que foram positivos, apresentados.

Com este sistema, o desenvolvimento torna-se mais económico, rápido e flexível e, uma vez que não depende do conhecimento de uma linguagem de programação, acessível para o público em geral.

Palavras-chave: Ferramentas de Desenvolvimento; Aplicações para Dispositivos Móveis; Realidade Aumentada; Visitas a Espaços; Interação; Jogos

Contents

List of Figures

List of Tables

Listings

CHAPTER 1

INTRODUCTION

We visit spaces daily. When we think of visits to a space, the first ideas that occur are the traditional visits to museums, art galleries or gardens but we are also visiting spaces when we go to the supermarket, to the restaurant or just by sitting at the dentist's waiting room.

These spaces are not meant to be visited in the same way as museums or art galleries, that have points of interest clearly defined. While in museums we can take a guided tour to find out more information about the space and its contents, in these spaces other forms of visiting need to be created. With the creation of virtual characters and objects we can bring the visiting experience of these spaces closer to the visiting experience of a museum or art gallery and create games and other interactive applications to help visit all kinds of spaces.

Whether the space is a waiting room, a restaurant, a garden or a museum, one of the main concerns of the managers and curators of these spaces is how to attract visitors and, once they are there, how to keep them interested and entertained in order to provide a good visiting experience.

Most of the main permanent collections and points of interest (POI) in museums and public spaces stay unchanged for a long time and in some cases do not ever change. To bring some dynamism and to encourage new visits many of these spaces offer temporary exhibitions that change every couple of months. Nevertheless, these spaces are far from having the deserved attention from the general public. Advertising is not enough and people need to be encouraged to visit them.

To enhance the visiting experience, some spaces can resort to visiting aids. Waiting rooms usually have magazines available to keep visitors entertained. Restaurants have little means of keeping a visitor amused while waiting for a table but some provide placemats with jigsaws and puzzles to be completed while waiting for the order. Spaces like museums and gardens usually provide maps and paper guides and, in the largest ones, audio guides can also be found. These means of providing information are useful but the amount and type of information is limited [Cha+14]. The audio guides are a popular solution but they can be boring and tiring due to the concentration needed to listen to them especially if there is background noise. It is also difficult to navigate through the audio. A more attractive way of communicating information is needed.

Most spaces offer web sites but they solely provide more information about activities and event advertisements. Some like, for example, the Louvre Museum, which created a virtual museum in Second Life [ML] and online 3D virtual visits [Lou], innovated but these tend to stay unchanged through time and allow little interaction with the space and other visitors.

People are more technologically aware than ever and spend a lot of time navigating the Internet searching for new information to learn or amuse themselves [Vis]. As such, we can take advantage of users desire to explore new technologies to develop new forms of visiting places and attract them, physically, to those spaces.

Visitors do not want to be overloaded with information, but to acquire relevant information, learn and have an interesting experience [Kuf+11]. A fast way to provide additional information is by using the visitor's personal mobile devices, that can be used to guide a visit, just provide extra information or be a tool to interact with the space and other visitors.

Augmented Reality can be used to give additional high detailed information about the environment [Mul+12; Rom+04; Wag+10]. With the creation of virtual objects and virtual characters, the space can be enlarged [SC12]. Virtual points of interest can be created. Virtual objects and characters can provide additional information that might be available in other spaces (for example, in other exhibitions) to complement the existing information.

Nowadays, new forms of user interaction are emerging that can lead to alternative ways of exploring spaces. Interactive applications that allow interaction with the space by manipulating objects present at the space or by performing gestures either with body parts or a mobile device are potential tools to help explore a space [Bea+07; PA14; Rub+15; SW07; Wag03; Wag+05; Woo+04].

The two main reasons why these interactive applications are not widely disseminated are the lack of personnel with the experience needed to create (or solely to update or maintain) them and the difficulty of installing extra hardware and objects (like tags or markers) at the space. This sometimes cannot be done because artefacts and environment are protected heritage and cannot be altered or just because of budget. The return on investment of these applications is no longer attractive as it is stated in the state-of-the-art review for developing serious games for cultural heritage in [And+10]. There is the need to create alternative solutions for the development of these applications that are cost effective, that do not require altering the space and that can be adapted to spaces with different characteristics, such as restaurants and museums. One way of achieving this goal is by using video camera networks, that can be often found at these spaces.

For the last 40 years we have watched video surveillance systems proliferate. Currently, the Closed-Circuit Television (CCTV) is often found in day-to-day life [HT04]. CCTV is

mostly used as a repository of video that serves three purposes: (1) provide images to a human operator to detect abnormal events or threats and take action, (2) capture information to be stored so that it can serve as evidence in the future or (3) is used in environments where conditions are dangerous to humans. Besides these purposes, the streams of video can be re-used to provide means for the creation of new applications.
With the evolution of technology it has become more affordable and easy to install video camera networks even in family homes. This evolution gave way to the development of automatic processing algorithms used for triggering of automatic events, which made surveillance quality increase and that can be applied in other areas of research.

Video Surveillance systems have evolved in three generations [Bra+06]. In the first, all equipment was analog. The video cameras captured information, which was sent to a central system that would archive it using tapes. In the second generation, information was still captured by analog cameras but information was transmitted to servers, which would digitize it. Today we are facing the third generation of video surveillance systems. Information is captured by digital cameras, which transmit the video streams through local computer networks to local servers.
In the last years these digital video cameras evolved into smart-cameras, which have an internal processing capacity. This allows compression and some processing of the information captured before transmission, improving transmission speed.

Nowadays, we can find the sign "Smile, you are being filmed." in most spaces, like museums, restaurants, supermarkets and shopping centres, meaning that a video camera network is present and filming. Even if there is not one present, a small network is easy to set up. By using these video camera streams we can recognise what is happening at a given room of a space, like determining which users are present and what they are doing. Gestures performed by the user's can be recognised and used as a form of interaction. Knowing what actions a user is performing at a given location of the space [Ana+06; Bea+07], provides means for generating information according to the user's context.

The use of wireless networks makes it possible for the information captured by the video camera network and the information generated to be transmitted to mobile devices. This information, which can be filtered by areas of interest, can be related to the space and users and combined with video from existing cameras, providing means for the user to explore the space and obtain more information about it.

Communicating additional information about the surrounding environment to the user and providing a way of interaction with it encourages the creation of new devices and applications for visiting these spaces. The computational support of an infrastructure composed of such networks of computers and cameras enables the development of applications for mobile devices that use it to perform computations which would not be possible to implement given their limited computational capacity and storage.

An infrastructure that takes advantage of video camera networks, that may be present at a space, together with a wireless network and the visitor's mobile devices provides means for the creation of interactive applications that help a user explore a space or just for entertainment [SC10].

1.1 Research Goals and Questions

The main goal of this research is to propose an infrastructure that enables the creation of interactive augmented reality applications to help explore spaces, like museums, gardens, schools, restaurants or even shopping centres. The objective of the visits will vary according to the type of the space and users. The infrastructure has to be capable of dealing with the different characteristics of the spaces (for example, indoor or outdoor).

This infrastructure has also to address the main problems found by space managers and curators for having these applications available at their spaces, namely, a user with little or no experience in programming has to be able to create and maintain the applications and not installing intrusive hardware at the space that could damage artefacts or alter the environment.

By not installing intrusive hardware and objects at the space we plan to use video camera networks and wireless networks to recognise the user and her actions and be able to generate relevant information that is communicated to the user's mobile device.

With the use of augmented reality we can create virtual characters and objects that will augment the space. Interaction with virtual characters and objects provides means to create interactive experiences. We aim at studying if this form of interaction leads to better visiting experiences.

From our research goal three research questions can be asked:

- **Research Question 1**: How to interact with virtual characters and objects present at a space?

  We aim at exploring how to interact with virtual characters and objects as a form of help to explore a space.
  Since our requirements involve using video camera networks and we will reuse the respective video streams to recognise the visitor's location and movements, interaction using gestures can be also recognised by processing the video streams. We will study if gestures are a good form of interaction and how gestures can be performed and recognised in order to create interactive experiences successfully. We will also study interaction with virtual characters and objects using the mobile device's sensors and input capabilities.

- **Research Question 2**: How can an end-user with little or no programming experience design and create augmented reality applications to help explore a space?

  One of the main reasons for mobile interactive applications that help explore spaces not being more wide spread is that its creation involves specialised human resources

capable of programming them. This makes the return on investment of these applications not attractive [And+10]. When the investment is made, applications tend to remain the same over large periods of time and are usually generic and not adapted to the user's interests.
If a space manager, a curator or some staff member would have means for designing, creating and maintaining interactive applications then affordable applications adapted to the age, interests and visit objectives could exist. For situations like, for example, a day where a visit of nine grade students studying History, a visit of nine grade students studying Art and a group of primary school children were expected, three specific applications could be created.
Visiting a space could be an experience closer to the visitor's likes and that could change over time. A good visiting experience is the best way to attract former and new visitors to the space.

- **Research Question 3**: How to create an infrastructure to augment a space and create interactive experiences, taking advantage of a video camera network without adding other objects or hardware?

  Since we do not wish to alter the spaces (some may be protected heritage) by adding objects to it, we need to work with what it is available.
  Most of these spaces have two characteristics in common: they have installed video camera networks and wireless networks. The requirements of the infrastructure is to have both these networks available.

  We were inspired by some projects. Project inStory [Cor+05], takes place at Quinta da Regaleira, located at the UNESCO protected Cultural Landscape of Sintra. It aims at developing a platform for mobile storytelling, information access, and gaming activities. With inStory a visitor explores the space while playing a storytelling game and visualising photos taken by other visitors. In this project a complex wireless network had to be carefully installed. This network was used, not only to provide means for client server communication, but to provide information about the user location.

  Tate Britain [Leh+07] recently employed two innovative systems in one of their major exhibitions of John Constable's work: a gestural interface and a touch-screen panel, both connected to large projection screens. Visitors could interact with the system by waving or touching the touch-screen panel and additional information would be visualised at the projection screens.

  In Milan's renown La Scala opera theatre, numerous innovative technological solutions were adopted for the exhibit: "Puccini Set Designer" such as 3D construction of sets, 3D character animation and costume animation, immersive Cinema, interactive documentary presentation and holograms [Spa04]. These solutions required the use of several rooms and the installation of numerous equipment.

In 2011, The Metropolitan Museum of Art took down the signs prohibiting the use of cellphones and created mobile versions of their website encouraging people to use it to browse the artworks [Loh] using their wireless network.
Museums have welcomed digital technology as not only inevitable but also as an ally in bringing culture to a wider public. There is the concern of how to encourage more "look up" experiences - up at the art instead of down at a smartphone or tablet. The Cooper Hewitt, Smithsonian Design Museum has undergone a massive renovation that increased the exhibition space and transformed the Carnegie Mansion into a 21st-century museum with several interactive experiences like an interactive dance floor, interactive wallpapers and tangible objects [Loh]. This museum banned the use of audio guides so that visitors could concentrate on the interactive experiences.

These projects were implemented at major museums as well as at protected heritage sites and even there it was possible to add hardware and objects. The requirements of this infrastructure are much more limited than the ones in the projects described above. Having this in mind we assume that, in spaces where these networks can not be found, we can install them and if the cameras of the video camera networks prove not to be sufficient in covering all the relevant angles, we assume that extra cameras can be added to the video camera network. We aim at not adding any other hardware, tags or objects.
By only having a video camera network and a wireless network as requirements and by not being a technological challenge or prohibitively expensive to install them if needed, the infrastructure can be adapted to a space provided that these two networks are present.

To answer these questions it will be necessary to design and implement an infrastructure that provides means for creating mobile interactive applications that use gestures as a form of interaction with virtual characters and objects as well as means to augment a space. This infrastructure can be adapted to a space, having as requirements a video camera network and a wireless network present or that can be installed.

There is a considerable amount of potential applications, including information systems for the general public, culture, entertainment, education, surveillance, indoor and outdoor activities and advertising.

1.2 Research Overview

In order to achieve the proposed goal, several solutions were implemented, experimented and tested. These solutions involved researching techniques and algorithms from several areas of research, such as Computer Graphics, Computer Vision, Augmented Reality and Human-Computer Interaction.

The proposed solution in this dissertation is a set consisting of a grammar, a visual tool and a modular infrastructure, which are described in Chapter 3. With the visual tool, applications to help explore spaces can be designed. An end-user with little or no programming experience is able to use the visual tool, designing applications easily and fast. This visual tool generates an application description file that complies with the grammar created. With this description file and the use of the infrastructure, applications can be created automatically.

The initial research was centred on the detection, tracking and recognition of people and objects and also on the recognition of people's actions. Many of the techniques and algorithms research are used in video surveillance systems and can be used to develop other kinds of applications. The research was conducted to understand the needs and limitations of these systems and to find the best candidates to be used in the infrastructure. Another task performed was the research of system and network requirements to perform streaming of the augmented video to mobile devices. This task proved to be one of the most time-consuming. An example of an application using augmented reality in mobile devices is described in Section 4.1.

It is important to know what these systems involve in terms of hardware and software to design and implement the modules of the infrastructure correctly. These modules can use several algorithms and techniques simultaneously to obtain better results. As technology evolves, these modules can be upgraded or re-written to accommodate new algorithms and techniques. A key aspect of the infrastructure is the definition of the input and output of each module, which makes the infrastructure flexible. To build applications all the modules can be used or just a few of them.

To showcase the capabilities of this system, five applications were implemented as working proofs-of-concept (see Chapter 4 and Chapter 3). Their main goal is to explore how interaction with virtual characters and objects can occur and to test the infrastructure modules. By studying how a person can interact with a virtual character or a virtual object we were able to adjust, correct and enhance the infrastructure's modules in order to accommodate these kind of interactions.

The main novelty of the proposed solution is the ability of an end-user with little or no knowledge of programming to be able to design and create augmented reality applications to help explore spaces in a few minutes (see Chapter 3). These applications allow interaction with the elements at the space using gestures.

1.3 Contributions and Publications

The contributions of this research work consist of an infrastructure, a grammar, a visual tool and five working proof-of-concept applications. Detailed contributions are:

- vuSpot: an infrastructure adaptable to a space that uses a video camera network and a wireless network to augment a space and provide means for interaction.

- XploreGrammar: a Domain-Specific Language (DSL) grammar to describe applications to help explore a space and where interaction is done by performing gestures.
- XploreBuilder: a visual tool to design applications for mobile devices to help explore a space and where interaction with the objects present at he space is done by performing gestures.
- urSpace: a proof-of-concept application built on top of vuSpot and that uses the XploreGrammar and the XploreBuilder applications to design and create mobile device applications to help explore a space. User tests and evaluation are presented.
- UBI, the Guardian Dragon: a proof-of-concept application built on top of vuSpot that augments the video stream of a space with a virtual caracter and provides means to interact with it using a mobile device. User tests and evaluation are presented.
- MagicLight: a proof-of-concept application built on top of vuSpot and allows interaction with virtual characters using mobile devices and pico-projectors. User tests and evaluation are presented.
- Haunted House and Gone Fishing proof-of-concept applications: in these applications we explore social interaction by inducing it with the use of mobile devices and pico-projectors to interact with virtual characters and objects.

Peer-reviewed publications:

Rossana Santos, Nuno Correia, "UBI, The Guardian Dragon: Your Virtual Sidekick", ACE '12, 9th International Conference on Advances in Computer Entertainment, **Full Paper**, Kathmandu, Nepal, Novembro 2012.
doi: 10.1007/978-3-642-34292-9_26
http://link.springer.com/chapter/10.1007%2F978-3-642-34292-9_26

Rossana Santos, Nuno Correia, "vuSpot: an infrastructure for augmented information sharing using surveillance cameras", ICDSC '10: International Conference on Smart Distributed Cameras 2010, PhD Forum, **Best Student Paper Award**, Atlanta, USA, 31st August to 4th September 2010.
doi: 10.1145/1865987.1866025
http://dl.acm.org/citation.cfm?doid=1865987.1866025

In addition to these publications a poster was submitted to the International Conference on Advances in Computer Entertainment (ACE 2015), a full paper was submitted to the IEEE International Symposium on Multimedia (IEEE ISM 2015), a full paper was submitted to the International Conference on Mobile Computing, Applications, and Services (MobiCASE 2015) and a full paper was submitted to the International Conference on Mobile and Ubiquitous Multimedia (MUM 2015). All are currently under review.

1.4 Document Structure

This document has the following structure:

- Introduction: the main motivations of this work, the problem statement, proposed solutions and contributions.
- Related Context and State of the Art: related work and a description of the state-of-the-art solutions and projects that influence this research.
- Creating Interactive Experiences: description of the infrastructure and tools created to achieve the research goal and answer the research questions.
- Exploring Interaction While Visiting Spaces: description and discussion of five proof of concept applications that were used to explore and study how interaction with virtual characters and objects can be done. This study presents solutions to the questions addressed in this research.
- Conclusions and Future Work: conclusions drawn and proposals for future work.

CHAPTER 2

Research Context and State of the Art

This chapter presents a review of the background and state-of-the-art of the relevant areas of research context and their applications: exploring spaces, visualisation challenges, mobile projection systems, video camera networks applications, gesture recognition and grammars and visual tools.

2.1 Exploring Spaces

Providing means for better exploring spaces is one of the main aims of this research. One of the biggest concerns is how to deliver to the visitor additional information about the space while providing a good visiting experience. Having a good visiting experience means that the visitor is more likely to come back to the space or advertise it to other potential visitors. The most common way of delivering additional information about a space is by having digital aids available. Exploring spaces can be more educational with the support of a digital aid that can communicate all kinds of traditional information, such as explanatory textual labels, posters, educational programs or other printed materials. Digital aids can range from a simple audio guide to a more complex multimedia mobile application.

The first systems using mobile technology to provide extra information while exploring a space used Personal Digital Assistants. One of the earliest projects (1997) concerning spatially aware devices is Cyberguide [Abo+97]. Cyberguide is a mobile, hand-held context-aware tour guide for the visitors of Atlanta's Georgia Tech University campus. The ideas behind Cyberguide created a revolution around digital tour guides and are still put in practice nowadays. They suggest gathering information about the visitors interests and patterns of visits to create a more reliable context and provide information adapted to the visitor's interests. It uses infrared beacons to determine the user's location indoors and GPS for outdoors. They concluded that the position information obtained is not reliable enough and to enhance it computer vision techniques should be combined. They also concluded that augmented reality is a potential technology to be used when visiting spaces and predicted that there would be a big evolution in this area of research.

Sotto Voce, is a guidebook designed to support social interaction between visitors and their friends [Gri+02]. A Personal Digital Assistant is used to display walls of a room. Each wall image has exhibition objects identified. By tapping on the object visitors can obtain additional information about it. Navigation between walls is done by using a navigation button.

The PortableCicero [LP04] allows the user to receive more information than an audio guide but due to the fact that they have no connections with other devices and no context-aware capabilities, information is kept on the device and it is the user's responsibility to navigate through it. This simplifies the hardware needs and makes the application lighter and easier to program.

Many systems adopted this strategy, where information about the current location of the user has to be given to the digital aid in order for the user to obtain the information relative to what is in view. This location information can be a number or a tag shown next to a point of interest that has to be inserted or a marker that has to be read. The use of markers for recognising the location of the user and points of interest is common and allows the creation of mobile augmented reality applications for educational exhibits [SW07; Wag03; Wag+05; Woo+04]. The project NaviCam [RN95] was pioneer on using markers and video cameras. The user carries a Personal Digital Assistant with a built-in camera to capture on video colourful markers placed on points of interest in the environment. When a marker is recognised, the user obtains more information about that point of interest.

Another technology that is used to detect the user's location is the Radio-frequency identification (RFID). RFID tags are becoming a popular form of identifying points of interest and have the advantage of being detected by the mobile device allowing the user to obtain the relevant information for her location without having to insert or read any information [Fev+11; Ghi+09; HF05; Hua+11; Mod+09; Sen+14; Tes+08].

In [HB05] a computer-enhanced study room is described. In this study room children could find a desk with a map of the exhibition and RFID-tagged keycards representing four objects in the Hunt Museum's permanent collection. The desk could detect the location of the cards on the map and provide the visitor with information about the object on the card, relating to the location on the map on which the card had been placed. This project has found that encouraging children to interpret artefacts for themselves neutralises the fact of children being normally voiceless while visiting museums.

In our research we do not intend to alter the environment. The proposed system has to be adaptable to the space and provide the functionalities without adding objects or hardware to the space. We intend to use streams from a video camera network that has to be present at the space to locate the user and recognise her actions. One technique that can be used when it is not desirable to alter the environment is recognising the natural features present at the space to detect and locate the user [Ong+05]. Another technique called visual recognition, in [Wei14], is to photograph the point of interest (for example, a painting). The application recognises that point of interest and is able to provide additional information. A usability study was conducted to determine if this technique was more efficient than using markers (in this case QR Codes and number codes). The study proved that users prefer to take photos of the point of interests then to insert numbers or read codes to obtain additional information about a point of interest.

We intend to use existing video camera networks present at the space. The only modification of the environment we are willing to make is to improve the video camera network by adding depth cameras or extra cameras to the circuit. Other research works use video streams to recognise the user and the user's location and their ideas and techniques can be used in this research. The POLYMNIA project [Ana+06; Bea+07] goal is to develop a platform for entertainment in leisure parks. Users visiting the park, register themselves and provide a face and full body photograph. With the use of cameras scattered throughout the park, visitors are tracked using cloth and facial recognition techniques. At the end of the visit it is possible to produce a video of the visit by combining the video footage where the visitor was recognised. These techniques can be used to locate a user in a space without the need for the user digital aid to possess a location system.

One popular use for mobile devices is the digital tourist guide. Tourists want access to information fast and easy and interfaces have to become smarter in order to meet user expectations. In [Ohs+03] a tour guide for thematic parks was created. This application offered short movies, games and additional information useful for the visitors, having an excellent visitor response.

The behaviours and circulation paths of visitors using a multimedia guide to help explore an exhibition were compared against the ones of visitors not using any support [Lan+13]. Visitors using the multimedia guide took longer to explore the space, spending more time at the points of interest of the exhibition and stayed longer at the space.

Knowledge acquisition and retention if favoured by playing a game compared to reading text because the player has to focus more strongly on problems [Bel+13].

A recent survey [Mor+14] has pointed out that the most common approach in digital games to explore spaces where visitors have points of interest (cultural heritage sites) are quizzes and puzzles, especially when played on mobile devices. Riddles and enigmas are also adequate for task-based learning [Ghi+09].

"Gossip at palace" is a location-base mobile game created to enhance cultural experiences inside museums [Rub+15]. Objectives of this work include: get an insight on the effectiveness of the game in facilitating the acquisition of historical knowledge and to measure the players degree of use of the digital game throughout the visit compared to analogous patterns registered for people using a multimedia mobile guide. They concluded that it offers significant learning potential and that the game facilitated a wider exploration of the museum. It proved to be more effective than a multimedia mobile guide to stimulate users' curiosity, inducing users to explore even the less prominent exhibit areas.

By comparing the learning performance while exploring a space between a user with an augmented reality mobile application and a user with an audio guide or no support, Chang et al. [Cha+14] concluded that the visitors using the augmented reality mobile application registered a better learning performance.

Exploring a space can be more efficient if the information available is adapted to the interests of the visitor, moving from a mass-oriented approach to a personalised experience approach [Abo+97; Mor+14]. GUIDE [Che+00] and Hippie [OS00] are location-aware electronic tourist guides. In these projects information to be delivered to the users is generated according to the user's characteristics and location. In [Kab13] a review of personalisation systems specialised for cultural tourism is presented. Visitors tend to visit the museum in small groups so technology should also facilitate group interaction [Kuf+11].

2.2 Visualisation Challenges

We aim at using the visitor's mobile devices to run applications that help explore a space. Understanding how to display information on a small screen and the techniques of displaying information to help explore the environment around us are key factors in creating successful applications for mobile devices that help explore spaces.
In this section, a survey of techniques and approaches to the visualisation of information in mobile devices is presented. The conclusions taken from the development of mobile applications to explore spaces or to guide the user can be applied to our research.

The exploring experience is influenced by the visual design of the applications. In [RK13] a study is conducted to assess the effects of visual style, information access selectivity and content-related challenge. They concluded that visual richness and added-content-related challenge positively affect the visiting experience, while the effect of information access selectivity was negative.

In [Fro+08] three case studies are presented exploring the challenges of visualising information for mobile spatial applications.

Together with the increase in power, computational capacities and functionalities of mobile devices, the evolution (and miniaturisation) of Global Positioning System receivers (GPS), wireless networks adapters, accelerometers and several sensors has forced a revolution on how information is displayed and manipulated on such devices.

One popular use for mobile devices is the digital tourist guide. Tourists want access to information fast and easy and interfaces have to become smarter in order to meet user's expectations. In [Ohs+03] a tour guide for thematic parks was created. This application offered short movies, games and additional information useful for the visitors, having an excellent visitor's response.

Digital tourist guides should always provide more information than printed guides and, most importantly, updated information. Many require an extra effort from the user because information held in the device must be first synchronised through a PC with

Internet or Wi-Fi spot instead of using real dynamically updated information delivered by a wireless network at the moment, which would be the ideal solution. Many researchers work on ways to make this information available on the moment and how to display it to the user with heterogeneous origins and formats of data. Mobile devices are now powerful computing devices but still with small interfaces and limited input capabilities due to the small displays and keypads.

In 1996, when the Cyberguide project [Abo+97] was created, mobile devices did not have the sensors nor the size and capacities that they have today. The Cyberguide's user interface is very simple and rudimentary but at the time it brought several advances in the way of displaying information, for example, the use of maps to show information about locations.

In [Bai+05] new types of interaction and visualisation were explored. The sensors that can be used together with the device allow it to be used in ways that were not possible before; for example, navigational buttons are now obsolete because the use of accelerometers and gyroscopes provide new forms of navigation and querying.

In 1998, Max Engenhofer [Ege99] introduced the term Spatial Information Appliance (SIA) to describe portable Geographical Information Systems (GIS) and described their future. In his point of view, GIS would start to be used by the general public on day-to-day use. They would have specific objectives and functionalities according to people's needs and device constraints. This customisation would bring a revolution in the way that SIG were thought and how the information was visualised and manipulated.

His vision of the future included the use of Geo-sketch pads, Smart-compasses, Smart-horizons, Geo-wands and Smart-glasses. These ideas are still put to practice and their combination provides a powerful basis for interaction and visualisation.

Geo-sketch pads are used, for example, to take field notes over photographs. A photograph is taken and it can be annotated with audio or text. The coordinates of the location are obtained from a GPS connected to the mobile device and attached to the photo. This way, the images of the real world are captured and complemented with the user own opinions and sensations providing a better context to the information captured.

In [Per+03], a more elaborated system, but that also uses annotations, is described. GeoNotes is a system that allows users to share personal annotations about locations, objects and everything that can be commented, promoting social-awareness. Annotating takes place in public spaces, which are selected using menus (the user does not need to have a GPS). The objects of those spaces are listed and annotations can be written. Users also have a profile, which means that information is filtered according to her preferences reducing the amount, and increasing the relevance of information to show the user. The way information is shown is very simple and has few menus but due to the input limitations of some devices it can be hard to place annotations.

In the Urban Tapestries Project [Lan03] a similar approach was used but in this case all interaction is done on a map. The users walk around a particular area of the city of London and are able to upload through Wi-Fi networks their annotations to a server.

These annotations can be text, images (captured by the user or drawn) and audio. One of the main aims of this project is to create "sound maps" of the city. Audio is considered one of the main triggers for memories. A user, while visiting the city, records audio clips and uploads them to the server with their respective coordinates. The concept of layer is introduced. Users have different likes and dislikes and a user may create a layer where her recordings will be attached. When putting all these sounds (or layers of sounds) together a tapestry of sound is created and it can describe the city in an innovative way.

Geo-Notes and Urban Tapestries have the same goal: to attach information to real world locations and objects. These ideas can be used, for example, to create social interactions when visiting a space or to create digital souvenirs.

Smart-compasses are compasses that show more than the north direction. They also show the direction of points of interest (POI) surrounding the user. This is a very simple and easy to use way of providing the user complementary information about his location.

POIs can sometimes be hidden behind buildings or outside the view range of the user. By pointing with the device on one direction, Smart-horizons indicate points of interest that the user is not able to see on that direction.

Geo-wands work in a similar way as Smart-horizons. The user points with the device in some direction and is able to visualise information about the POI she sees on that direction (or nearest POI in that direction). This is a different and easy way of making requests to a database or GIS.

Smart-glasses are used in Augmented Reality (AR) systems. The user sees the real world through the glasses and, with the combined information retrieved from a GPS and a gyroscope; information is visualised over the lenses superimposing the reality (for example, to see the flow of a dry river on a ravine or to get an historical flashback of an artefact).

The visualisation of information in each individual case can be quite straightforward but when combining these approaches information and more complex interfaces can be created. This evolution of technology and demands gave way to the Mobile Spatial Interaction (MSI) [Fr09]. MSI requires much more from the applications than spatially aware systems usually do. It requires a sense of orientation and interaction between the application and the user's physical surroundings to relate the locations in the user's real world and the digital information. This interaction is only well accepted by the user if the design and usability of the interface is appropriate, being visualisation one of the biggest concerns in these systems.

When designing MSI applications some design dimensions have to be taken into account:

- *Orientation-awareness:* it is important to know how users use their cognitive functions to match the real world with the virtual that they visualise on the mobile display. Visualisation of the virtual world can be more efficient if the user can find easily and quickly her actual location in the digital map. This is the starting point for navigation and search for information. The most usual and traditional ways of organising information like menus and listings can be used to search for information but due to the difficulty that the smaller devices have in providing user input, new forms of selecting information have been thought.

Application interfaces can have different designs [Fr06]; for example, information can be searched using pointing gestures (by pointing at a location, and pressing a button, information about points of interest in the direction pointed can be viewed), maps (by selecting or tapping, in the case of nowadays PDAs and some mobile phones, the location on a map), radars or augmented reality. The pointing gesture was proven a very efficient and attractive mean of searching for information [Fr06].

A map can be oriented north (i.e., the north of the map is always displayed on the top of the display) or oriented by what the user is seeing (this is called "egocentric" view). In this last case, what we visualise on the display refers to what we visualise in front of us (providing that the device is held in front of us and upright). Orientation-awareness describes the ability of mobile devices to rotate the display in order to align the virtual world with the real one.

In [Her+03] was conducted a study to determine whether egocentric or north-oriented maps are the best to help the user navigating. They show that egocentric maps supported the navigation best. Many users, without noticing it, rotate physically the devices. This can induce errors. Egocentric maps are the ones where less user navigational errors occurred.

- *Representation of environmental objects*, like buildings, can be shown in several ways, for example, as a 2D billboard model or a 3D block model. Photo realistic models are the best accepted and desired but the small displays and low computational capabilities of mobile devices make it difficult to render very detailed models. Also information has to be filtered. In applications that allow user input, especially in the form of annotations, it is important to have some filter mechanism of the information to visualise or it can be distracting, overwhelming and also not into the users interests.

- *Perspective:* several types of perspectives exist, for example, first-person view, bird's eye and vertical.

  When given more than one target, users tend to achieve one at a time because the visualisation of the whole area is difficult. The use of the vertical view to fly over the area is often used and helps in the task of placing oneself on a location [Her+03]. Many times users apply a simplification: they choose the "bird's eye" point of view or the vertical view, i.e., viewing the map from the sky. This makes it possible to look at a 3D map as a 2D map.

- *Field of view:* the field of view depends on how much of the surrounding virtual world should be shown. Users should have the freedom to choose the field of view they want in order to define the amount of detail they want to see.

  In [Fr06] a study was held in order to understand what type of views users prefer to use. The conclusions show that users appreciate the use of mobile phones as tour guides assistants and prefer views that relate real POIs with modelled 3D objects rather than to using coordinates of locations. The traditional menus and listings were also well accepted. Users that are not comfortable with reading maps preferred the Augmented Reality approach. It is a popular approach but has the problem of possible information overload. It is especially useful for remote viewing (i.e., viewing POIs that are not in the view of the user).

The Point-to-Discover Project (p2d) [Fro+08] describes a mobile service from which users can get information about restaurants in an area of $3km^2$ in the inner city district of Vienna, Austria. This project was done in collaboration between the mobile phone operator Mobilkom Austria and Siemens Austria, a mobile service developer, so the mobile devices used are regular mobile phones. From pointing with the device at a restaurant, the information (and respective recommendation) available for that restaurant is visualised. The ideas first formulated by Engenhofer [Ege99] are enhanced and refined to develop a virtual pointer to real world information. By using a field of view detection algorithm, the information visualised is always related with what the user is actually seeing of the real world.

The solo use of GPS devices to obtain the orientation of the user's movement was proven not to be enough [Bai+05]. Orientation can be found while moving but when the user stops deriving the orientation is impossible. The interface design is very important in this case. Users have to quickly know where they are in order for the application to be successful. In printed maps this is done many times by turning the map physically and holding it facing the direction of the target. It is also easily achieved with 2D digital maps by rotating the device. For this, electronic compasses became a must-have item in guiding systems. Without them users must depend on memory, observation and orientation skills to position themselves in the map.

In [Sim+06] the SIA concept introduced by Egenhofer was enlarged to address the functionality needs of smart, spatially aware personal geographical assistants. All that a person can see, hear or write can be referenced to a location in the form of annotations just like a post-it or wall graffiti. The different platforms that exist make it difficult to design an application that fits all. The devices characteristics vary and they might have several limitations like low computational power, small displays and limited user input capabilities. These limitations force researchers and developers to use different kinds of visualisation techniques to display the different kinds of information. Also the data structure has to able to support information for various scenarios.

In the "Creative Histories - the Josefsplatz Experience" Project [Bai+05; Tob; Fro+08], an orientation-based human interface was developed. The project proposes to show the user an architectural environment of a square in Vienna through time (the device display would work as a "window to the past") based on a 3D model, created by joining several photographs and images of the locations.

Viewer applications were created both for desktop and mobile devices. One of the main concerns in this project was the simplification of information in order to be possible to visualise information in devices with very small displays. With the Creative Histories mobile application the user can see how the points of interest, namely the Austrian National Library, were in the past decades along with other information (speech, text, audio and video) that can be visualised by selecting icons rendered on the 3D model. The information that appears on the mobile device is chosen according to the users orientation so that information to visualise refers to the locations just in front of the user.

This application is of great help when sightseeing, the user is spared from the task of finding information and has access to information in ways not possible with ordinary printed tourist guides, like video and sound. Augmented reality can be used to superimpose information over what the user is seeing. This can be achieved by using a camera along with the mobile device. Nowadays most devices have a built-in camera so new forms of Augmented Reality are emerging. When visualising information that is not visible (i.e., in the view of the user) different approaches are possible [Fr06], for example, map, map with visibility differentiation (for example, use different colours for visible or not visible POIs), radar, radar with visibility differentiation, Google Earth, Google Earth with visibility differentiation , and Augmented Reality (seeing "through" buildings by accessing photos of the ones behind).

When using 2D maps with mobile devices, users rely on street name information and other readable information like street crossings and green areas. Pre-knowledge of the area also reduces the cognitive effort. In Figure 2.1 a table of cues and their frequencies based on data from 75 tasks is shown [Oul+09].

The cues that users use to orient themselves are non-overlapping but the strategies used to read them are the same as reading traditional printed maps and the simplifications the human mind does are similar. This data is important when designing the functionalities of the applications and interfaces so that users can find easily the information they need to facilitate their orienteering. This is not what happens with 3D maps [Oul+09]. The referential relationship between the virtual world and the real world might not be easily achieved and it is more difficult to navigate between locations. In the study conducted, people found it more difficult to walk to a target and they did it slower than when using 2D maps. When using 3D maps, people tend to shift to 2D-like strategies for orientation.

Nevertheless, 3D maps have several promising qualities. They allow: (1) a detailed representation of the cityscape by using volumes to represent the relief, (2) multiple views, being the first-person view the most common, (2) realistic details, which can not be put into a 2D map, for example, POIs can be represented with very rich detail imitating the

Cue type	3D	2D
Known landmarks	Very often	Often
Building shapes	Often	–
Facade details	Often	–
Facades (whole)	Often	–
Relative directions	Often	Sometimes
Street names	Often	Very often
Street crossings	Sometimes	Often
Blocks, or part of blocks	Rarely	Sometimes
Parallelism of streets	Rarely	Rarely
Cardinal directions	Very rarely	Sometimes
Store/office names	–	Rarely
Street number	–	Rarely

Figure 2.1: Catalogue of cues [Oul+09].

real world so they can be easily recognised and (4) several degrees of freedom in the users movements on the map, making it possible for the user to explore the map as she would explore the real world and finding hidden details that can be looked up afterwards in the real world. In 3D maps, the egocentric view is the most commonly used point of view found. It allows looking directly at the target without having to first mentally infer its direction. Doing this with 3D maps requires the use of a digital compass because the orientation of the user is not known. Users prefer to use a combination of 2D and 3D maps getting the most out of each qualities. The use of a 2D map to give a general, non-detailed, view of the surroundings is often useful for the user to choose the best path to take or just to be situated on the map. In [Oul+09], the authors point out that 2D maps are more effective than 3D maps. This is not surprising if we think that 2D city maps are products of decades of work. Also we are used to read maps and signs since we are small children and not everybody are 3D-gamers. 3D maps are very promising and allow new types of interaction (for example, for games and entertainment applications). More detailed information can be presented along with the usual information found in 2D or paper printed maps.

Augmented Reality systems are about 40 years old [Azu+97; Mil+95; Val98]. The evolution of the hardware, especially for graphical processing, made these systems evolve rapidly in the last decade. Games and other commercial applications were developed and with smartphones becoming popular mobile applications emerged [Mul+12; Wag+10].

Augmenting a space can be done by using markers, recognising features or by using location information obtained from sensors. In this research the approach used was to obtain the location from sensors or from processing video camera streams.

An Augmented Reality system for the visualisation of the underground infrastructure is presented in [Sch+09a]. With a hand-held video camera real world images are captured and information about the underground structure is superimposed guiding the user over the infrastructure.

The client is equipped with a 3G card (that enables communication with the server), GPS and video camera. The location of the user is obtained using the GPS and a query is sent to the server. With the coordinates of the user, the server calculates a 3D model of the scene, which is represented by a XML document that is sent back to the client. The client is now able to reconstruct the scene with the captured video and the information received.

In Project ANTS [Rom+04], an Augmented Reality technological infrastructure was developed that can be used to explore physical and natural structures, namely for environmental management purposes. It had two prototypes, the 1998 Lisbon World Exposition (EXPO 98) venue application and the water quality monitoring application. The main applications for this system are: monitoring water quality levels in natural water bodies and artificial lakes, visualisation of the characteristics and temporal evolution of physical structures and natural elements by the superimposition of synthetic images of the past or predicted scenes on real images and superimposition of synthetic images on the ground to reveal the soil's composition at the user's current spatial location (for example, the location of underground water supply networks and subsoil structure). The information is superimposed using the Bubble Metaphor [TP12], which means that the size of the bubble corresponds to the importance of the information displayed.

A head-mounted display was used to visualise the real world with information superimposed. For virtual and augmented environments, it is essential that the equipment is the least intrusive possible and wireless [Ras+06]. Nowadays, head-mounted display are non-intrusive enough to be used in commercial augmented reality systems.
When combined with digital compasses and gyroscopes, these devices provide the user a way of looking at the world that does not require much cognitive effort from the user. Still, any quest for information has to be done using another device like a PDA. The final developments of the project allow users to use a PDA instead of a head-mounted display and most of the techniques used can be applied to create space exploring applications.

In the WikiVienna Project [Fro+08] the work done in p2d and Creative Histories is combined. With the use of server-side technologies to render and reconstruct the views, it is possible to bring Augmented Reality to mobile phones. The application was developed to reach the mass public and for that reason interaction and visualisation are the key roles for getting the best experience out of the device. In WikiVienna a whole block was fully 3D modelled and can be explored as in opposition to the p2d project, where buildings were modelled with blocks and cylinders.

Algorithms to dynamically insert virtual elements in images where used in [NC13]. This algorithms can be used to augment video frames with information in a way that the information does not collide with the space structure and contents.

Modelling the space from the video stream frame images [Nob11] is a strategy to ensure that, when superimposing information, it is placed in the image at a position where an object does not exist.

A device-aware adaptation, which is transparent to the user, is made in order to deliver the right amount of necessary polygons for a smother rendering on each different mobile

phone. Also a content adaptation is made. This allows for the file size and quality of the generated images to be adapted to network bandwidth and mobile device capabilities. WikiVienna allows the user to express herself through the use of annotations. To attach annotations to locations, the user searches with the device keyboard the buildings or objects and can magnify them to choose the precise location where the annotation should be placed. These actions guarantee that the location of the annotation is precise. If using GPS coordinates, the GPS distance error would make annotations to be placed in wrong objects since in an urban environment objects are very close to one another. The 3D model was constructed based on a simplified model provided by the city of Vienna. Reconstruction of the model is collaborative. Users upload images that are later used for the model reconstruction. Since users use cheap and low-resolution cameras, the images uploaded have different quality and sometimes reconstruction of the buildings is not satisfactorily achieved. To overcome this problem, better reconstruction algorithms are under development.
Augmented Reality improves the access to the POIs and its identification. It also opens new doors for visualisation and interaction.
Using projectors along mobile devices solves many of the problems related to the small display of the mobile device. This gives the freedom to create applications that deliver complex information, which was impossible before. The next section describes the state-of-the-art of mobile projections system.

2.3 Mobile Projection Systems

In the last few years we have watched projectors evolve and be reduced to sizes that allow them to be integrated in mobile devices. Mobile devices with projection functionality started to be commercialised [Sam].
Projecting technology has greatly evolved and handheld projectors, called pico projectors, can be bought at accessible prices. Many of these pico projectors can be attached to mobile devices through cables [Mice].
The problem of the small displays in mobile devices can be minimised using projections as displays [GR09; Han+08; Sch+09b]. By using a pico projector attached to a mobile phone, its display is projected providing a large area to visualise media, overcoming the output limitations of the mobile phone. They also give way to the creation of new forms of interaction. We intend to explore how to interact with virtual characters and objects using pico projectors.

We aim at using the visitors' mobile device. The device may have projection functionalities of a pico projector attached but we wish not to make any other hardware required. The following projects give insights on the potential that handheld projection has and on what information can be displayed and how.

A desktop metaphor was used in [Win+11], where a user interface that provides access to the mobile device content and provides a space to share content with other call participants. A projection is used to enable the use of the mobile device functionalities while making a phone call.

The implications of using projections to display information and how people will want to use such technology is explored in [Wil+12]. They outline recommendations about design and privacy. The results obtained indicate that projecting information is appealing to people and may solve mobile information needs.

The Magic Lens metaphor [Kaw+10] is created using a pico projector attached to a mobile phone and mobile augmented reality to interact with smart objects. By processing the video stream of the mobile device, camera tags from objects are recognised and information about them is projected. Tasks like searching books or sorting boxes can be carried out easily due to the projection.

The MITs SixthSense [MM09] and Wear Ur World [Mis+09] are wearable gestural interfaces that augment surfaces near the user with digital information. They capture a video stream of what the user is viewing and augment the objects around letting the user use natural hand gestures to interact with that information. They showed the potential that projection applications have in augmenting the world around us.

In [Win+14] this idea is further explored with Ambient Mobile Pervasive Display (AMP-D), a system that provides a vision of pervasive ambient information. Relevant information from a mobile device is projected on the floor while the user is on the move. When private information needs to be displayed, like a text message, the user can use her hand to project it. The user interacts with the system using hand gestures. With AMP-D a separation of the public and private information was created allowing the user to use the system as an extension of the mobile device.

REFLCT [Kru+12] is a mixed reality projection framework that uses a pico projector on a head mounted display to project computer generated images. This system is to be used by militaries, law-enforcement authorities or in any other domains that benefit from a simulation system that combines computer-generated imagery with physical tools. Virtual characters are created to give the user additional information about the simulation.

As virtual characters can be an extension of the owner's personality, the personality characteristics could be obtained from social profiles as well as the user's input. A personal visualisation projection system is evaluated in [Leu+11]. The idea behind this system is that a user social identity is different than her real life identity and, by projecting complementary personality characteristics obtained from social profiles, the user's online and offline presences become closer.

The Videowalker project [Bon12] aims to address the urban space and the natural environment by projecting contextual audiovisual material. It explores how video material through a personal projection becomes a human output modality potentially enabling new forms of audiovisual expression.

An Imaginary Friend that is projected on the floor and walks along with the user is described in [RC11]. Using a sensor to measure the electrodermal activity of the user, the Imaginary Friend knows the emotions that the human companion is feeling and reacts according to them. The concept of using electrodermal activity to identify emotions in real time could be used to make virtual characters emotions match to the owner's emotions.

Disney Research has several projects using pico projectors. These projects have very interesting results, but they need a specific mobile device built especially for them, which is not the aim of this research. Nevertheless, the interaction techniques they use give guidelines on how we can interact with virtual characters using handheld projectors.
They use invisible fiducial markers in the projection to indicate where and what it is being projected. In project SideBySide [Wil+11] the markers indicate the position of characters. In project HideOut [Wil+13] the markers are used to map projected imagery onto tangible objects and surfaces such as storybooks and board games. These projects do not require instrumentation of the environment since the markers are projected in the near-infrared (IR) spectrum becoming invisible. The devices they use are equipped with video cameras that film the projections and markers. These video streams are processed and marker information extracted the new information to project is generated.

2.4 Video Camera Networks Applications

By reusing existing video camera networks, this research intends to provide alternative solutions for common problems in mobile applications that help explore spaces, such as determining the location and actions of a user. By knowing the user's location and what she is doing, the information to be displayed can be adapted to the user context.
To obtain the user's information the video streams have to be processed and people and actions have to be recognised. Many of the conclusions and algorithms used in video surveillance systems can be applied to obtain this information.

The real-time detection of threats using security systems like CCTV (closed-circuit television) requires, from a trained human operator, a level of attention that hardly can be maintained for long periods of time. The area of automatic video processing has been gaining importance for security and detection of threats (for example, monitoring a motorway through the detection of immobilised vehicles, cargo and wrong-way drivers [Bra+04], smoke detection [Vez+08] and identification of people in airports [Li+14]).

Much of the technology associated with the detection of threats can be applied in other areas of research. The algorithms used in automatic video surveillance can be used in the creation of virtual or augmented environments when used to recognise people and their actions at a space.

In Video Flashlights [Saw+02] is presented a system that uses video streams from several cameras and maps them as textures, in real-time, on a 3D model.

One of the main difficulties in implementing systems that use distributed video networks is the detection of objects and people and determining their location. The techniques used for automatic surveillance, for example, identification of people and objects [Wei+14], their location and movement [Akh+14; Lis+14; Rin+07], facial and body expression [Vio+04; Zha+03a] and actions [Zha+14] can be used for other applications.

Distributed video systems solve some problems of individual camera systems, such as the concealment of objects, but introduce some new challenges as the detection of elements in motion between spaces [Rin+07].

Another major challenge faced in this type of systems is the recognition of the space. Real world conditions, such as the weather, influence the results of processing the video streams of the space. The IBM People Vision System [Con+04] proposes solutions for some of these issues, namely the identification of individuals and objects moving on moving scenarios.

The use of video equipment is becoming increasingly more accessible, often at the expense of purchasing low-resolution video capture equipment which poses difficulties for the recognition of objects and the space. On the other hand, with the new generation of cameras (smart, depth and stereo cameras), some information processing can be done in the hardware of the camera, namely compression, making communication more efficient and depth information can be obtained. With these new technologies recognition and tracking of users is done faster and more accurately [Vez+13].

In [Bra+06] high-performance embedded cameras that combine video sensing, video processing, and communication within a single device are described.These smart cameras are key components for novel surveillance systems. The UMass Smart Space [Ou+04] combined virtual reality with real video information that is handled locally by the smart-camera, which then communicates the necessary information to the system of virtual reality to mirror the real world. Speed is gained as the bandwidth used is much smaller and the transition between cameras is facilitated.

A possible use for the captured video information is its enrichment with information relevant to the user or the context, which is chosen according to location, profile and activities. This augmented video information can be delivered to the user in the form of a guided tour or used within a game to help explore a space.

In [Azu+01] is presented a survey of technologies used in Augmented Reality and its applications. A video-conferencing system is proposed in [Pri+02]. The virtual viewpoint system for the video-conferencing is also used to generate live 3D avatars for collaborative work in virtual environments, for example, a guide in a virtual art gallery.

Pervasive games can be played using augmented video streams in mobile devices. The Project IPerG [Ipe], from the Mixed Reality Lab (Nottingham), explores how to provide a high quality interactive experience for pervasive games. Project ANTS [Rom+04] implemented an Augmented Reality technological infrastructure that can be used to explore physical and natural structures with mobile devices that display augmented information.

Transmitting a video stream and displaying it in heterogeneous mobile devices imply its restructuring. In [Pha+04] is discussed the problem of restructuring the information for heterogeneous mobile clients. The use of mobile devices for viewing and manipulating video requires an increased computational capability that most mobile devices do not have. A solution to this problem is to search, in the vicinity of the device, for other suitable machines and use them, opportunistically, to perform the computation instead of the mobile device [Bal+07; Yan+08].

The use of the information acquired is a controversial issue even when used by law enforcement agencies and other entities for crime control and it is often alleged to breach the privacy of citizens [HT04]. This is an issue that may limit the use of the information and has to be carefully thought.

2.5 Gesture Recognition and Grammars

Performing gestures can be a form of interaction used in the system purposed in this dissertation. In our research, gestures (actions) are captured on video that is processed in order to extract the information about what the user is doing. The information to deliver is generated according to the user's actions.

Gesture recognition is widely used in sign language recognition. The successful recognition of lexical signs is not enough for the correct understanding of the meaning and other non-manual signals need to be recognised, such as facial expressions, to provide context. Another difficulty in recognising signs is detecting transitions between signs in continuous signing. These issues are also present in systems where the user's actions need to be recognised. The issues of integrating non-manual signals and how to recognise them are addressed in a survey in [OR05].

The depth-mapping existing in Microsoft Kinect video cameras [Micb] can be used for sign language recognition and verification when playing educational games for deaf children [Zaf+11]. When comparing a Kinect-based system to CopyCat, a system where users need to use coloured gloves and embedded accelerometers, the Kinect-based system showed more potential being a viable option for sign language verification. By detecting signs more accurately, depth cameras are good candidates to also be used in whole body gesture recognition.

Gesture recognition can be used in other areas such as recognising aircraft handling signals [Son+11]. The NATOPS Aircraft Handling Signals Database need not only hand but whole body gesture recognition to successfully determine which signal is being communicated. The procedure of gesture recognition, from single acquisition to interpretation is described and can be applied to this research. Signals are communicated continuously from an unsegmented and unbounded input stream. Like in sign language analysis, detecting the beginning and end of a sign poses a major difficulty [Son+12].

The Etruscanning Project presents virtual reconstructions of cultural heritage sites inside museums [PA14]. The visitor is able to move around at the virtual site by performing gestures in front of a large display where the virtual site is being visualised. Making sure the visitor performs the gestures correctly is one of the goals of the project. They state that not being able to get the gestures recognised correctly is one of the main reasons for frustration while using this kind of systems. They aim at helping the visitor understand how to perform them correctly by using interactive tutorials.

Besides being simple to understand and have elegant representations, grammars are extremely popular in systems using gesture recognition because of their ability to model semantic representations of patterns. Various methods and applications that use grammars for solving inference problems in computer vision and pattern recognition are reviewed in [CD04].

Computer vision systems that use grammars to describe recognised human actions can be used as inspiration to our research.

Series of gestures can be seen as complex actions. With the creation of action units that form building blocks to model complex human activities, actions like common work in a kitchen, like preparing breakfast cereals, can be recognised [Kue+14] and a grammar is used to represent the human activities. Actions are inherently hierarchical by human nature and are modelled like words in a speech, being the human activities compared to sentences.

A detailed overview of various state-of-the-art research papers on human activity recognition is provided in [AR11]. In this overview the methodologies developed for simple human actions and those for high-level activities are discussed and an approach-based taxonomy is chosen that compares the advantages and limitations of each approach.

The results of a systematic review to identify the state of the art in the area of computer vision and pattern recognition using grammars is presented and discussed in [Ped+13]. They used a systematic review process to investigate the main digital libraries in the area and to document the phases of the study in order to allow the auditing and further investigation. Focus was given on learning or extracting the structure of the grammars from images, as well as those that use grammar-based approaches to perform image recognition. They also point out unexplored research opportunities in the literature.

An hierarchical approach for the recognition of high-level activities using grammars is proposed in [IB00]. Complex action sequences are recognised in video. They defined the structure of the sequences as a relationship of primitives. Their approach has two levels: low level primitives are recognised using a statistical approach and the higher level parses actions, which are configurations of primitives, using context free grammars. By encoding a large number of stochastic productions rules they are able to explain all activities possible in their environment. This approach was extended in [ME02] to recognise multitask activities. They showed through experiments the recognition of complex activities, such as, the identification of player strategies in multi-player card games, like blackjack, where actions like " a dealer dealt a card to a player" are recognised.

The segmentation problem of recognising multiple objects is addressed in [Min+03]. They implemented a system that uses human-specified grammars to recognise the activity of a person trying to solve the Towers of Hanoi mathematical game. They were able to recognise the inherent action of the person without information about the objects in use, depending solely on the motion information of the activity.

A context free grammar for attribute recognition was described in [JC06]. By attaching semantic tags and conditions to the production rules of the grammar, the recognition

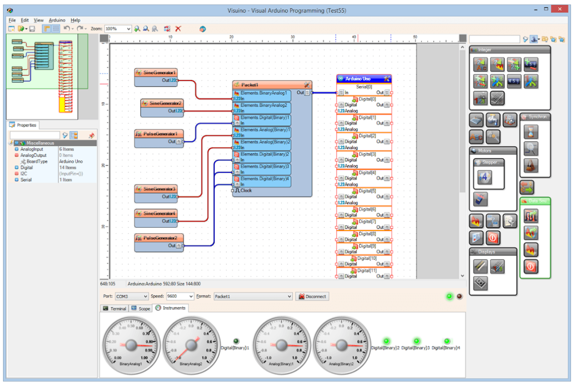


Figure 2.2: Visuino [Sof].

of more descriptive activities is possible. When observations satisfy the syntax of the grammar, the system decides an action occurred. Events in a parking lot can be recognised using this system. By representing the typical activity in a parking lot, it is possible to detect an abnormal activity.

The recognised gestures are not always performed by only one person. Actions can be recognisable gestures like approach, depart, point, shake hands, hug, punch, kick and push. A Context-Free Grammar was used in [RA06] to represent composite actions and interactions between of two persons.

Recognising accurately large sets of gestures is still an open research problem. As the gesture set size increases, the recognition accuracy will tend to decrease rapidly. Using video game context for game-based virtual environments can improve the recognition accuracy of large sets of gestures [TI+15]. By comparing similar games with and without context it was concluded that, when including context in full-body gesture recognition, a higher recognition accuracy is obtained allowing for more gestures to co-exist.

2.6 Visual Tools

Visual interactive development environments (IDEs) create new applications by moving programming instructions, building blocks, or code nodes to create documents that can be compiled or interpreted.

As Visuino by Mitov Software [Sof] a visual programming environment (see Figure 2.2) used to program Arduino [Ard] boards advertises: "For those people who are not strong on writing code then designing, compiling and creating Arduino programs has never been easier! Why waste time on creating code when we have done all the hard work for you already? You have your Arduino board, and great hardware design, see it running in minutes, not hours!", the main advantage of using visual tools is not having to memorise the language and details used to program the application. Developing applications becomes easier and fast.

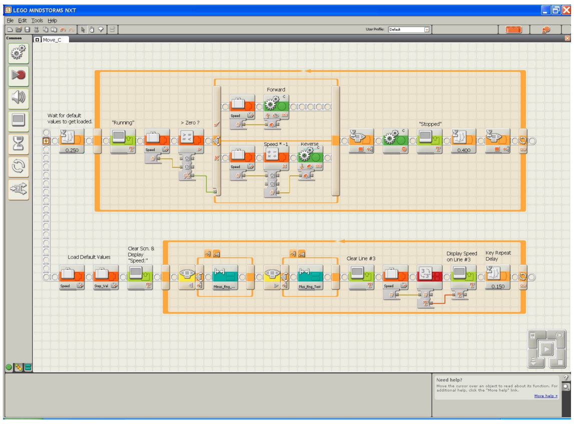


Figure 2.3: Lego Mindstorms [Leg].

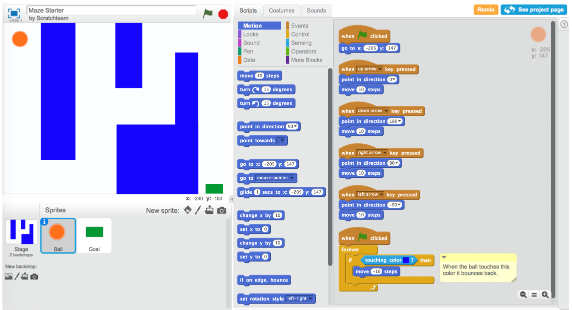


Figure 2.4: Scratch [MIT].

Many examples of visual tools to develop applications can be found but only a few are directed to end-user development. In our research, we are focused on creating a visual tool for end-user development. Visual tools for developers have a lot in common with visual tools for end-user development and many ideas and techniques can be applied.

The Lego Mindstorms [Leg] was a pioneer in creating a programming language that consisted of building blocks (see Figure 2.3).. Programs to command robots can be written by assembling these blocks. Thanks to the simplicity of the building blocks and rules of assembly, anyone over 10 years old can take a first step into programming, technology and creative design.

A programming language specially developed for children is MIT's Scratch [MIT] (see Figure 2.4). It is suitable for children of all ages due to its simplicity. Although being simple to use to the point of small children be able to use it, complex applications can be built by more experienced users. The complexity of the code depends on the building blocks used, their configuration and arrangement.
Many of the strategies for moving and assembling blocks and the design of the interface in the two applications can be applied to XploreBuilder.

There are also visual development tools dedicated to game development. Microsoft Kodu [Micc] is a visual programming language made specifically for creating games and provides specialised primitives derived from gaming scenarios (see Figure 2.5). It

Figure 2.5: Kodu [Micc].

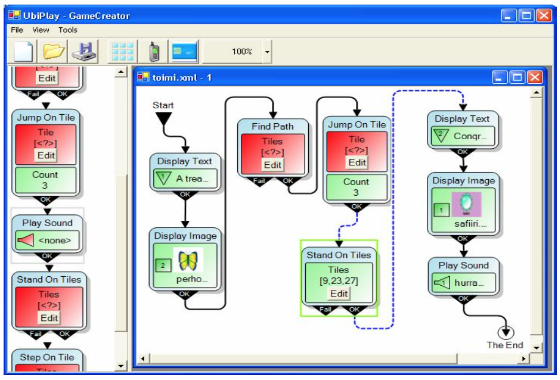


Figure 2.6: UBIPlay [MV06].

is designed to be accessible for children but can be used by anyone. Kodu can express advanced game design concepts in a simple, direct, and natural manner.

UBIPlay [MV06] is a technology platform for programmable interactive playgrounds. UbiPlay allows children to create and play games in interactive playground environments. (see Figure 2.6).

A visual tool to develop context-aware applications called iCAP is proposed in [Dey+06]. This tool requires no programming experience and uses rules and elements (objects and activities) to define the application prototypes (see Figure 2.7). The rules are composed of elements that are dragged onto them. Another application that does not require any programming experience and works with blocks that are assembled (like in our research) to create and configure sensor-based systems is described in [CV05]. This paper describes a series of experiments showing that non-expert users have much difficulty with a block based on Boolean logic truth tables, and that a logic block having a sentence-like structure with some configurable switches yields a better success rate. In our research, logic blocks are essential when designing complex interactions. The conclusions reached in this paper will help creating a better interface. The former two applications can

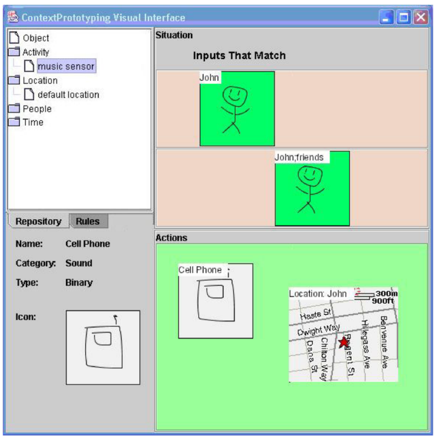


Figure 2.7: iCAP [Dey+06].

be used to create applications to explore a space and create tour guides and games but not for mobile devices.

Minigames, such as multiple-choice quizzes, puzzles or open-answer questions, can be created by teachers and museum educators in [Roj+14]. These mini-games can be played on the visitors' smartphone by installing a player-app. Having mini-games available to embed in the interaction experience client application is a functionality to consider.

Edutainment mobile applications can be developed using MADE [Bel+03]. Unlike in our research and in the previous systems, MADE is to be used by developers and is not suitable for the end-user development. In this paper the problems related to the interaction between the user and the mobile device when rich multimedia content is presented are address.

Closer to this research we can find Cicero Designer: An Environment for End-User Development of Multi-Device Museum Guides [Ghi+09]. Cicero Designer allows people without any programming experience to customise functionalities and the interface of mobile museum guides. Interaction with the user is done by using solely the input capabilities of the mobile device. While some ideas on how to design applications can be used it does not provide means for interacting with the space.

2.7 Telepresence

In 1999, Jim Gray enumerated the "Dozen Information-Technology Research Goals" [Gra03] and telepresence was considered an innovative way of producing high-quality experiences. Ten years later, in the 2009 ACM Multimedia Challengers [Acm], RadVision proposes the creation of "a meeting experience that equals or surpasses in-person meetings" showing that telepresence is still a very active research area.

Telepresence may include, vision, sound and manipulation of the environment. The goal is to simulate the human activities possible in a given environment so that the remote user can undergo similar experiences as a person on the location, consequently the system must be able to provide ways of reproducing these activities remotely.

There are several well-known applications for telepresence, for example, teleconferencing, e-learning, work done in environments that may put life at risk and remote surgery.

Steve Mann [Man09] has been researching, since the 80's, how to create wearable computers. He wishes to show the world exactly as he sees it by using a wearable wireless webcam, called "WearCam". Using the Internet, he displays real-time video captured by the WearCam and visitors of the web page can see what he is seeing. The ideas behind the development of this device can be explored and used in other forms of telepresence, namely when using mobile devices' cameras to give guided tours of a space.

In the project TOURBOT - Interactive Museum Telepresence Through Robotic Avatars [Rou+01; Tra+00], the telepresence concept is explored. In this project, a museum can be visited remotely by giving commands through the web to a robot at the museum. This robot will film the environment surrounding it and the user can see remotely that environment as if being there. The robot is also able to provide local personalised guided tours.

Telepresence can also be used to enlarge the information available for the visit. In the Carnegie Museum of Natural History a colony of insects can be visited remotely [AN01]. A robot is placed in the insect colony and visitors to the museum can remotely observe the insects in their real environment, providing a very educational experience. The video image is scaled to human size and the insect sounds amplified, giving the impression of being inside the colony and being the same size as the insects.

2.8 Summary

In our research we propose an infrastructure adaptable to a space to provide means to create interactive experiences to help explore a space, a language grammar to describe interactive experiences, a visual tool to design interactive experiences and an application to create interactive experiences created on top of the infrastructure.

Mobile devices are a privileged tool to help explore a space. Most of the applications used to help explore spaces are mobile. The first created were audio guides but as soon as they were commercialised, Personal Digital Assistants were used [Abo+97; Gri+02]. As technology evolved applications became more complex, visually attractive and with more functionalities [LP04]. The functionalities in these applications can give hints on what the visitor needs when visiting a space and can be used when creating interactive experiences to help a visit.

Knowing the location of the visitor is one of the main concerns of these systems, in order to provide the correct information to the visitor [Che+00; OS00]. Location can be determined with the use of markers [RN95; SW07; Wag03; Wag+05; Woo+04], RFID tags [Fev+11; Ghi+09; HB05; HF05; Hua+11; Mod+09; Sen+14; Tes+08], by recognising the visitor in a video stream [Ana+06; Bea+07; Ohs+03] or by recognising elements in the

space by filming them and processing the video stream [Ong+05].
In our research we aim at recognising the visitor by using the streams from a video camera network present at a space [Azu+01; Bra+06; Con+04; Rin+07; Saw+02; Vez+13]. Video surveillance techniques can be used to process the video streams [Akh+14; Bra+04; Li+14; Lis+14; Rin+07; Vez+08; Vio+04; Wei+14; Zha+14; Zha+03a].

Studies show that it is a success to use mobile device applications like games and quizzes [Mor+14; Rub+15] to guide the visitors through a space and that it improves the learning potential [Bel+13; Cha+14; Lan+13]. Interactive experiences can be based on games and quizzes.

Augmented reality is a popular technology in this kind of applications and provides means to transmit additional information to the visitor. Most of the visualisation solutions found in applications [Azu+97; Bai+05; Ege99; Fro+08; Fr06; Fr09; Mil+95; Mul+12; Oul+09; Per+03; RK13; Rom+04; Val98; Wag+10] to help explore spaces can be taken into consideration in our research.

The small display of the mobile devices can pose a problem when visualising complex applications [GR09; Han+08; Sch+09b]. The use of pico projectors minimises this problem allowing applications to have better and more detailed interfaces [Bon12; Kaw+10; MM09; Wil+13; Wil+11; Wil+12; Win+11; Win+14].
In our research we intend to explore the use of handheld projectors as a mean of interaction with the space.

To be able to provide means for a user with little or no experience of programming to create interactive experiences applications, a form of describing the space and interactions needs to be provided. Interaction with the space can be done using body gestures [OR05; PA14; Son+12; TI+15; Zaf+11]. A common way of describing this kind of interactions is with the use of a grammar [CD04; Kue+14; RA06].

The description of the space and the interactions at the space can be generate by an application that is a visual tool. Several examples of visual tools used to program applications can be found [Ard; Bel+03; CV05; Dey+06; Ghi+09; Leg; MV06; Micc; Micd; MIT; Roj+14]. The structure of the building blocks, their organisation, actions to assemble them and application layout can be taken into considerations when creating this visual tool to design interactions.

Chapter 3

Creating Interactive Experiences

One of the main goals of our research is to propose an infrastructure, adaptable to a space, that by only using video camera networks and a wireless network provides means for augmenting a space, creating interactive applications to help explore it. To achieve this goal we created vuSpot, an infrastructure with a client-server architecture that is described in the next section (Section 3.1).

In vuSpot, on the client side, we use mobile devices to run the interactive experiences' applications to help explore a space. Another goal is to provide means for the creation of mobile device applications without the need for the user, who is creating the application, to write its source code, which means the user is not required to know how to use a programming language.

We created the urSpace application, which is described in Section 3.4, to create interactive experiences. To create interactive experience applications, urSpace needs a description of the space an of the interactive experience. In Section 3.2 we propose a Domain Specific Language (DSL) grammar called XploreDescription, which is used to describe interactive applications to help explore spaces.

Writing a description file based on a grammar is not an attractive task and the user need not to know what a grammar is or what it represents if a visual tool to write the description file is provided. In Section 3.3 we describe the XploreBuilder, a visual tool for designing applications. This tool generates an application description file that complies with the grammar proposed and a space description file. urSpace is able to create the interactive experience application given these files.

With these four components (vuSpot, XploreDescription, XploreBuilder and urSpace) we are able to design and create interactive experiences visually and in a simple way. These interactive experiences are based on gestures made with mobile device or on gestures performed by the user that are captured on video by the video camera network. By using this strategy, we do not need to add objects or hardware to the space. We can use this four components at a space provided that there is a video camera network and a wireless network available, which means that it can be adapted to almost any space that fulfils these requirements.

3.1 Supporting Infrastructure

vuSpot [SC10] is an infrastructure, adaptable to a space (see Section 3.1.1) that uses existing video cameras networks to provide means for augmenting spaces and support interactive experiences.

It was implemented in C++ using the openFrameworks framework, which allows the use of the OpenCV library and several video processing addons. The algorithms used can be replaced for more efficient ones as technology evolves. It is not the objective of this research to develop such algorithms but to propose an infrastructure that combines them in order to be possible to create applications to explore spaces that can take advantage of their strengths.

vuSpot consists of a set of modules that can be combined, being structured as a client-server architecture (see Sections 3.1.2 and 3.1.3). By adopting a client-server architecture, it is possible to delegate on the server side most of the computations. Some of these computations, such as the video processing, would be difficult or impossible to perform on a mobile device due to the limited processing and memory resources. A description of the server and mobile client architectures is presented in the next sections.

The adoption of a client-server architecture also makes it possible to support heterogeneous clients. Both client and server are composed with modules that can be used to assemble different configurations of the system according with the space characteristics (for example, indoor or outdoor), target audience and desired interaction experience.

The client is any mobile device that has the ability to communicate with the server using a wireless network adapter and has the ability to perform computations (for example, smart phones, Personal Digital Assistants and custom devices).

This infrastructure must have the ability to deal with client differences in terms of technology. The client may have several sensors (for example, gyroscope, digital compass, GPS, camera and microphone), some form of haptic feedback like, for example, a vibration motor and a video projector instead, or together with, a common LCD display.

Nowadays, most mobile devices have built-in or attached video cameras. The video streams from these cameras can be used to recognise other users, objects or provide a new view angle of the space [Man09; Pha+04].

By taking advantage of the computational capacities of the computers connected to the video camera network, the video streams, both from the video camera network and the mobile devices, can be augmented and transmitted to the mobile devices of the local visitors.

For the video streams to be visualised by the users with the least delay possible, equipment [Ras+06] and streaming techniques have to be carefully chosen. Transmitting video streams at an acceptable rate is a problem of low-bandwidth networks. It can be minimised using smart-cameras [Ou+04; Rin+07], which have an internal processing capacity and can perform simple computations and compression of the video streams before sending them to the server.

Not all applications use the video from the video camera network in their client application (examples can be found in Sections 4.2 and 4.3). In these cases, the captured video is used solely to recognise elements and actions in the video streams to generate information to be displayed by the mobile device. The applications that use video streaming to the mobile device are the most demanding in terms of technology.

The infrastructure is prepared to be used in many scenarios (see examples of applications in Chapter 4). In the less demanding scenarios modules can be used only in part or not used at all.

In the space to visit, several elements exist, for example, points of interest, virtual characters and objects and visitors. These elements have to be recognised in the video streams. The points of interest are stationary physical locations that exist at the space and contain information that is relevant for visiting it. Virtual characters and objects can be added to the space to augment it and can only be seen by those using an application created to help visit that space. These virtual elements have the ability to move.

The visitors location can be obtained using the information supplied by the user mobile device (GPS in outdoor spaces), Wi-Fi triangulation or from the video stream of cameras present at the space. Visitors are recognised using location [Com+y], body [Wre+97], face [Vio+04; Zha+03b] and cloth [Ana+06; Bea+07] detection algorithms. Objects, which might not be tied to a location have to be located using the video streams [Ong+05]. Other elements, like virtual characters and objects are shown to the user to augment the space. The augmented information and the video stream combination shown to the user had to adapt while moving around the space.

By watching the augmented video, users know which and where virtual elements are present at the space. With this information users are able to interact with them. Interaction can be in the form of simple gestures performed in the surroundings of an element. By analysing the video streams from the video camera network, gestures are recognised (with action recognition algorithms [Wre+97]) and the virtual element can react to those gestures. If a gesture is performed in the surroundings of a point of interest, the reaction to that gesture can be seen in the mobile device display. The reaction may be additional information related to that point of interest, superimposed onto the video stream of that location.

Other locations in the space can be visited. In this case the user will receive a video stream of that (probably not in the view range) location captured by the cameras present there.

Like in the real world, each object has specific characteristics and is manipulated according to its purpose. Virtual objects have to be designed considering how the manipulation will occur and their interface must be evaluated [Nor02].

The visitor can also interact with virtual characters that appear in the augmented video stream displayed by the mobile device. This form of interaction induces the creation of guided tours and games.

Visits to the space can be taken in the form of guided tours (that may contain interaction with the space), following a virtual character and receiving from it additional information about the space and points of interest. Additional virtual characters can be found along the tour to give emphasis to the context.

These guided tours have to be able to adapt to the changes of bearing of the user. A user can explore the space at her will. The video streams from the video camera network are used to detect where the space is most crowded and guide the user avoiding the crowds.

A more interactive way of visiting the space can be taken in the form of a game. In a game, beside the virtual information available for all visitors, other virtual characters and objects give hints that help the visitor to discover hidden details of the space and guide the visitor through the space in the quest for answers to enigmas.

Users can interact with each other using virtual objects and characters. Virtual objects can be collected and passed from user to user.

3.1.1 Adapting vuSpot to a Space

All computations performed by vuSpot depend on the processing of the video streams captured by the cameras of the video camera network present at the space and on the sensor and input data communicated by the client application using a wireless network that also needs to be present at the space.

The ability to recognise users and the gestures performed by them depends on the number and location of cameras. To obtain better results, video cameras have to be placed strategically to cover specific angles. When defining the elements of the space that can be used in the interactive experiences (such as, Points Of Interest) attention has to be given to the fact of whether the gestures performed near that element are captured successfully by the video cameras. If they are not captured successfully then either that element should not be considered for the interactive experiences or an additional camera should be added to the video camera network to capture that element and the users and actions surrounding it.

The best results are obtained when the visitors body is fully captured on video from the front and without objects between the camera and the visitor. Of course this is the ideal setup and it is not what is found in spaces where the video camera network is only used for surveillance. In this cases, the video cameras are usually placed at the corners of the rooms and most of the time the cameras will capture the visitor from behind, which is also an acceptable setup because gestures can still be recognised if not occluded by the visitor's body. A tutorial to be shown before using the application is shown to have made a difference in [PA14], as well as in the application in Section 4.2. In this application the visitor is captured on video from behind (see Figure 3.1). The users that had problems with the gesture recognition when facing the points of interest hanged on a wall (because they were performing the gesture in front of the body, meaning that their body was blocking the camera view), after a short tutorial on how to perform gestures had no problem with the gestures' recognition.

Problems arise when the visitor is captured sideways. For example, if a visitor looking at a Point of Interest will stand sideways in the camera view then the gestures that the visitor may perform may not be captured on the video streams because the visitor's body is blocking the view and it does not matter how the visitor performs them. Virtual Characters and objects should also be placed carefully at the space in order for the interaction using gestures to be successful.

If the video camera network present at a space is used solely for surveillance purposes, probably there will be the need to install additional video cameras on the circuit to capture specific angles of points of interest.

Nowadays not only big public spaces have wireless networks installed. No matter the use of the space (museums, restaurants and shopping centres have them available) wireless networks are likely to be found at spaces that people visit. If they do not exist, installing them is not a technological challenge or prohibitively expensive.

If there is not a video camera network present at a space then, in order for the infrastructure to be adapted to that space, it has to be installed. In this case, the best angles can be chosen and the cameras placed carefully so better results can be obtained.

In Figure 3.1, examples of spaces where video camera networks were installed can be observed. These examples correspond to the applications created in Chapter 4.

One of the aims of this research is not to alter the spaces (some may be protected heritage) by adding objects (tags, hardware or other objects) to it. The probability of finding these networks at protected heritage sites and major museums is high since they are used for surveillance purposes to check if the visitors harm the artefacts or steal them. In several projects [Cor+05; Leh+07; Loh; Spa04], implemented at major museums as well as at protected heritage sites, it was possible to add hardware and objects and even to alter the space configuration. The requirements of vuSpot are significantly less limited than the ones in these projects. We assume that, in spaces where these networks can not be found, we can install them and if the cameras of the video camera networks prove not to be sufficient in covering all the relevant angles, we assume that extra cameras can be added to the video camera network.

By not needing to install additional objects other than the ones needed for having a wireless network and a video camera network, the investment of installing vuSpot at a space may only be of installing vuSpot infrastructure software.

In the same way as in other systems that help explore a space [Cor+05], to configure

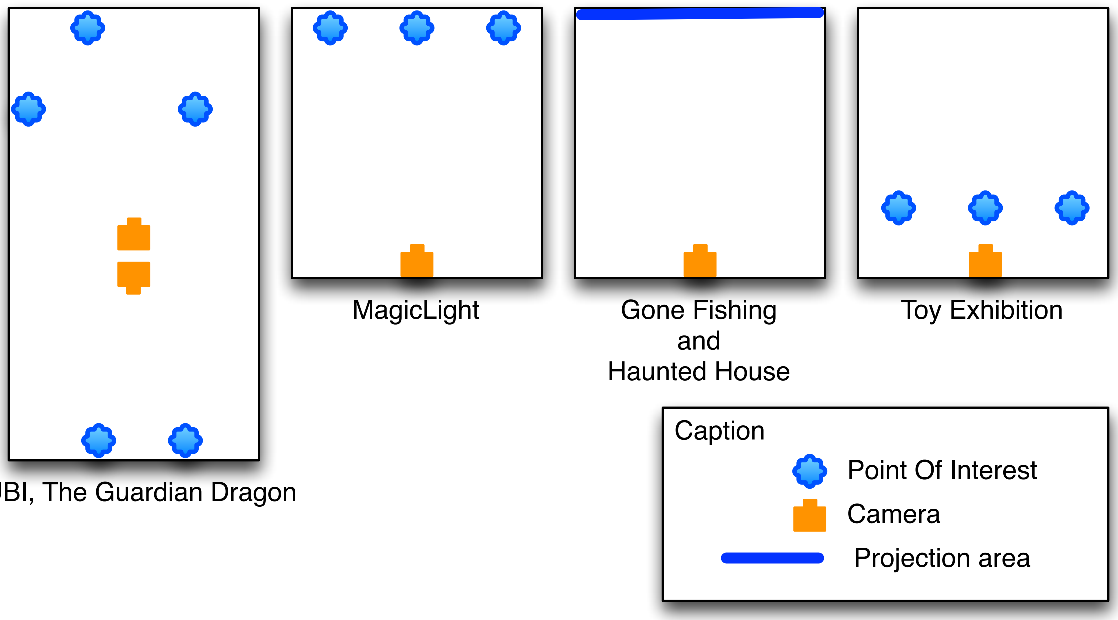


Figure 3.1: Adapting vuSpot to a space.

vuSpot, contents (such as, information about the points of interest of the space) and the information about the physical characteristics of the space are needed. Gathering this information is the most time consuming task but is only done once.

When a space meets the requirements, vuSpot is a fast and affordable infrastructure to install at a space compared with other systems that use tags or dedicated mobile devices.

3.1.2 vuSpot Server Architecture

The vuStop server is composed of several modules (see Figure 3.2). A module can be used as a whole, in part or not at all according to the mobile device application objective. All information (space definition, user information, interactive experiences information and other auxiliary information) is stored in a database. The server contains the following modules:

- Communications Manager: is responsible for all communication between the server and the clients. Earlier on this research we used the Transmission Control Protocol (TCP) to exchange information but soon we realised that communication was being delayed and that, since the information exchanged is composed of small sets of strings or are images, the User Datagram Protocol (UDP) had a better performance. The communications manager gathers all information needed from the infrastructure modules and sends it to the clients. It also receives information from the clients, such as video streams, sensor data, user input or information generated by the client application, and redirects it to the correct modules of the infrastructure to

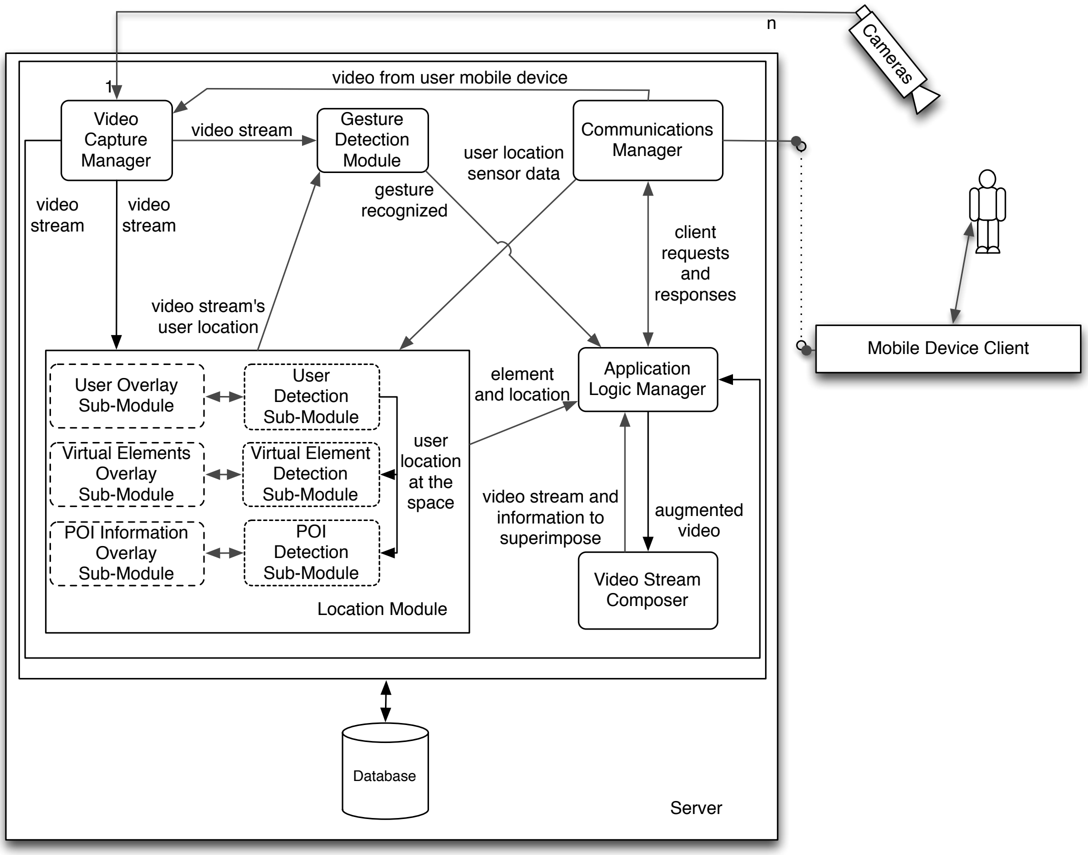


Figure 3.2: Server Arquitecture.

be handled. The information (apart from images and video) received and sent is written in eXtended Markup Language (XML). Information about the XML messages exchanged can be found in Section 3.1.3, where the client is described.

- Video Capture Manager: receives the video streams of the cameras present at the space. It manages the list of available cameras and respective reference to their video streams. Since several cameras are present at the space (at least one per room) an index of cameras and their location has to be maintained. The Detection Module, the Location Module and the Application Logic Manager can make requests for video streams.

- Location Module: is responsible for finding the location of all the elements present at the space: users, points of interest and virtual elements. Since locating each of these three elements uses different techniques, they were separated into sub-modules. It delivers the element identification and location information of the recognised elements to the Application Logic Manager. Users are tracked using the sensor data received from the client (normally GPS data), using blob tracking, face and cloth recognition algorithms and depth data. Points of interest and virtual elements information are stored at the database. The difference between them is that virtual elements are not fixed at the space and they can move. Their existence and location depends on the user location and stage of the visit.

- User Detection Sub-Module: one of the key aspects of the infrastructure is the finding the location of the user, either in the video stream frames (2D) and at the space (3D). This is the responsibility of the User Detection Module. The location of the user can be found using two strategies: from sensor data received from the client (in this case the user location and identification is direct) or by using the video streams of the cameras present at the space, and received from the Video Capture Module, to detect and recognise the user. In the second strategy, if depth information is available the resulting location of the user is more accurate.
  The performance of the system may increase by modelling the space from the video stream frame images [Nob11] or by using video surveillance algorithms [Akh+14; Li+14; Lis+14; Rin+07; Vio+04; Wei+14; Zha+14; Zha+03a] to locate and recognise the user at a space using the video stream.

- Virtual Elements Detection Sub-Module: detects virtual elements (objects and characters) in the surroundings of the user location at the space by performing requests to the database. If a virtual element is detected, information about it is sent to the Virtual Elements Overlay Module.

- POI Detection Sub-Module: detects real objects (points of interest) in the surroundings of the user location at the space by performing requests to the database. If a point of interest is detected it is sent to the POI Overlay Module.

- Overlay sub-Modules: with the information received by the corresponding Location Sub-Module, generates the information to overlay on the video stream (on which the user is captured). The information is the information relevant for the user visit, like information about points of interest in the space, virtual characters, virtual objects and user avatars. The Overlay sub-Modules generates the information's visualising location on current display area (if video is in use then the current display area is the current frame, otherwise it is the whole display area of the mobile device).
  The overlay sub-Modules can use information from algorithms to dynamically insert virtual elements in images [NC13] to better locate at the space video stream the element to overlay.

- Video Stream Composer: composes the information generated from the Location Module (element id to superimpose and location in the frame) and context information generated by the Application Logic Manager (for example, alerts and other messages) with the video stream where the user was detected. The augmentation, of the video stream where the user is in, is done frame by frame with the elements generated by the Overlay Sub-Modules of the Location Manager and context information obtained from the Application Logic Manager.

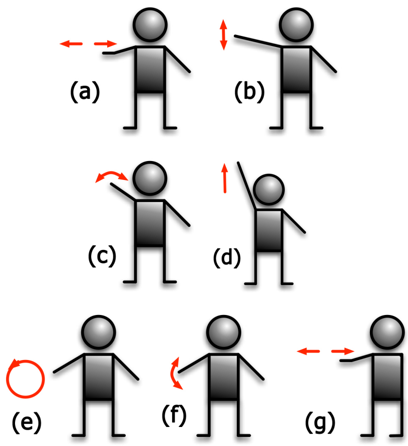


Figure 3.3: Examples of detected gestures: (a) poke, (b) tap, (c) wave, (d) raise hand, (e) circle, (f) shake and (g) click

Algorithms to dynamically insert virtual elements in images [NC13] can be used to augment the video frames. This augmented video is sent back to the Application Logic Manager. This module is only used in the case of augmented video streams are needed for the client application.

- Gesture Detection Module: Gestures are movements the user performs using the body. By knowing the user's location provided by the Location Module, the Gesture Detection Module processes the video stream where the user is located, detecting the movements the user is performing. Gestures like poke, tap, wave, raise hand, circle, shake hand and click can be detected (see Figure 3.3). The Gesture Detection Module returns the recognised gesture identification retrieved from the database. The detection of actions is achieved by calculating optical flow averages at specific locations (for example, user's arms and hands, using openFrameworks with OpenCV and the addon ofxCvOpticalFlowLK from Flightphase) or by using the OpenNI (Open Natural Interaction) Kinect Project. Other algorithms and techniques [Kue+14; RA06; Son+11] can be used in this module to enlarge the set of gestures recognised. As the gesture set size increases, the recognition accuracy will tend to decrease rapidly [TI+15]. Having a large set of gestures makes the performance of the system decrease and interactions might become excessively complex when combining gestures. The size of the gesture set should be just enough to allow the creation of interesting applications.

- Application Logic Manager: is responsible for generating information about the context, interpret user actions and generate the interaction results. It gathers information from the other modules of the infrastructure and makes requests to the database to perform computations that generate new information. It is responsible for updating the database with the gathered information and the new information generated.
  It also defines the expected behaviour of the applications, for example:

- Visualising Virtual elements: if the client application uses a projector, the virtual elements are sent for visualisation to the client application (see Section 3.1.3) who displays them on the screen display area. If not using a projector, they are not sent to the client for visualisation and the only way of visualising them is if the client application displays a video stream augmented with the elements.
- Projecting: in the case of applications created to be visualised using the mobile device display and when projecting them does not bring an added value (for example, the Toy Exhibition in Section 4.4), if a projector is connected and if the behaviour for when using a projector is not specified they will be visualised in the projection just as they would be on the device screen.
- Context information: interaction can depend on objects in possession, points of interest in the surroundings and past actions. It can be done just to gather information, to score points in a game or give hints to continue the visit.
  Interaction with points of interest (real or virtual) depends on the location of objects and object features. This information is retrieved from the database and together with the information about the allowed interactions, the reaction is generated.
  Other context information is generated, for example an element to superimpose when the user points with the mobile device projector outside the projection area, like in the application Gone Fishing in Section 4.3) or warning messages.
- Collections: as a result of the interaction, if collections are allowed at the space, we might keep the element, drop it, use it or perform other actions only possible with that object.

The generated information to be visualised on the user mobile device screen is sent to the Communications Manager to be delivered to the client application.

3.1.3 vuSpot Client Architecture

Figure 3.4 shows the mobile client architecture, which, as for the server, is divided in modules.

- Interface: allows the user to interact with the system. The mobile device can give haptic feedback to the user, display video streams and receive input from the user through the use of the mobile device input capacities. The interface can be projected by a pico projector attached to the mobile device or, if the mobile device has projection capabilities, through its own projector.

- Information Manager: is responsible for gathering information from the sensors and input from the user and create an eXtended Markup Language (XML) message with it to send to the server (see Listing 3.1) and for parsing the XML message received from the server (see Listing 3.2), obtaining the information to be visualised.

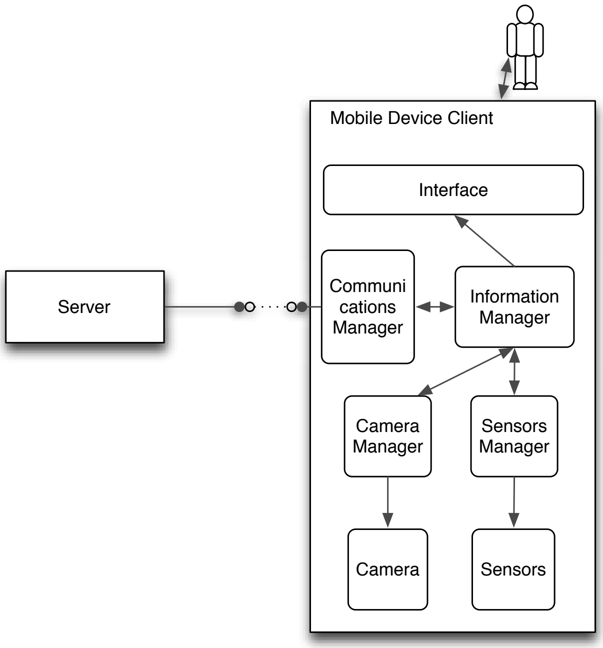


Figure 3.4: Client Arquitecture.

- Communications Manager: exchanges with the server all information needed by the system.
- Camera Manager: manages the device video camera stream (if any) and returns its reference to the information manager.
- Sensors Manager: manages sensors that might be present and returns the information gathered to the information manager.

To use the application the user needs to login into the system. The username and password are sent to the server encrypted with the Advanced Encryption System 256 (AES256). The server will respond with a message containing the user identification in case of a successful login or, otherwise, with an error message. After a successful login interactions and visualisation of the information transmitted may initiate.

The client application contains all the content to be visualised. It has two responsibilities: gathering sensor data and user input and sending it to the server, and visualisation of the information received from the server, which are described in the next sections.

3.1.3.1 Gathering sensor data and user input and sending it to the server

The information needed to perform computations by the vuSpot server from the user mobile device can be obtained from the accelerometer, gyroscope, compass and GPS sensors that can be present at the device and from the user input.

The client application obtains information from the sensors and user and creates a XML message with the information gathered to send to the server (see Figure 3.1).

```xml
<vuSpot>
  <userID>1</userID>
  <input>
    <DESTINATION>a</DESTINATION>
  </input>
  <accelerometer>
    <x>-0.730383</x>
    <y>-1.167121</y>
    <z>1.996781</z>
  </accelerometer>
  <gyroscope>
    <yaw>30.75</yaw>
    <pitch>50.34</pitch>
    <roll>10.01</roll>
  </acceleration>
  <compass>
    <TrueHeading>242.878</TrueHeading>
    <HeadingAcc>25</HeadingAcc>
    <x>38.0053</x>
    <y>-16.8948</y>
    <z>8.80006</z>
  </compass>
  <GPS>
    <latitude>38.6238</latitude>
    <longitude>-9.13461</longitude>
  </GPS>
  <projector> NO </projector>
  <camera>YES</camera>
</vuSpot>
```

Listing 3.1: XML message sent by the client.

The "userID" tag refers to the identification of the user in the system.

The tag "input" indicates if any input was entered by the user and if so, the respective variable name and value. In Listing 3.1 the variable "DESTINATION" has value "a". If there is no input then the "input" tag is empty.

In the "accelerometer" tag, the values for acceleration obtained from the accelerometer are three: x, y and z, corresponding to the axis of the movement.

In the "gyroscope" tag, the values for rotation obtained from the gyroscope are: yaw, pitch and roll and are given in degrees.

In the "compass" tag, the compass sensor information is composed of the true heading in degrees, heading accuracy, and (x,y,z) values of the direction of movement.

In the "GPS" tag, the GPS sensor returns information for the latitude and longitude of the user.

The server also needs to know if a projector and a camera are present. This is indicated in the tags "projector" and "camera" respectively. If any of the sensors described are not present at the mobile device then the corresponding information has -1 values. If there is no input information then the input tag is empty.

3.1.3.2 Visualisation of the information received from the server

The client receives an XML message containing the current information to display. It can also receive a video or an image depending on what to display.
The information in the XML message consists on the reactions to be visualised that were triggered by the user by performing action near a Point Of Interest and the environment that may need to be visualised like, for example, in applications that use a virtual scenery to be projected.

```
<vuSpot>
  <userID>1</userID>
  <POI>
    <id>2</id>
    <x>10</x>
    <y>0</y>
    <z>20</z>
    <orientation>centre</orientation>
  </POI>
  <POI>
    <id>3</id>
    <x>0</x>
    <y>30</y>
    <z>15</z>
    <orientation>centre</orientation>
  </POI>
  <TEXT>
    <id>INFO1</id>
  </TEXT>
  <MULTIMEDIA>
    <id>IMAGE1</id>
    <location>centre</location>
  </MULTIMEDIA>
  <MULTIMEDIA>
    <id>IMAGE2</id>
    <location>centre</location>
  </MULTIMEDIA>
  <INPUT>
    <id>INPUT1</id>
  </INPUT>
  <QUESTION>
    <id>QUESTION1</id>
  </QUESTION>
</vuSpot>
```

Listing 3.2: XML message sent by the server.

In Listing 3.2, examples of possible tags are shown. Not all applications use all tags. The "userID" tag refers to the identification of the user in the system. Several "POI" tags can follow. A "POI" tag indicates a virtual element with a given "id" to be visualised at coordinates (x,y,z) of the display and with a given orientation (left, centre or right). The "text" tag indicates a text with a given id to be display. The "multimedia" tag indicates an image, video or audio element with a given id to be displayed or played. In case of images, a location tag can be used to specify the location in the displayed area where the image is to be visualised. This is useful when using a pico projector to visualise the image and was used in the MagicLight application (see Section 4.2) for the dæmon to indicate a direction to the user. The "input" tag indicates a input with a given id to be asked to the

user. The "question" tag indicates a input with a given id to be asked to the user.
Visualisation of content has six possible layouts:

- Virtual Elements (POIs): are visualised at the position that was calculated by the server (they can move and possess an initial location, speed and direction of movement). Depending of the direction of movement the image may need to be inverted according to the its orientation.

- Text: is visualised using the whole screen. Text is aligned left, not justified and centred vertically. If the text takes more space to be visualised than the size of the display, a scroll bar is shown.

- Images: are visualised using the whole screen and will be stretched or reduced in order to fill the whole display area, maintaining aspect ratio.

- Videos: are visualised at the centre of the display and will be resized to fit the width of the display, maintaining aspect ratio.

- Questions: the question text is displayed on the top of the screen. If there is an image to display it will be visualised below the question. The image will be resized to fit the width of the display, maintaining aspect ratio. The multiple choice buttons will be displayed below the question text or image, if it exists. If the size of these components exceed the display area then a scroll bar will be shown.

- Input: an input request is visualised in the same way as a question.

3.2 Describing Applications

One of our goals is to provide means for the creation of interactive experiences to help explore spaces to people with little or no experience in programming.
As such, we need to provide an alternative mechanism to writing code. The user needs to have a form of describing the application she wants to create in detail. This description can be passed to an application generator (see urSpace in Section 3.4) that knows how to parse it and create the application immediately and without any actions from the user.
We use a language to describe applications that are interactive experiences to help explore a space. XploreDescription is a DSL grammar created to describe applications to help explore spaces (see rules in Listing 3.3). The grammar provides means for describing both simple and complex interactions. Interactions are described as sequences. These sequences are related to a location and have an action associated that causes a reaction. By being possible to write applications with both simple and complex sequences, the users can choose the level of complexity of the created applications according to their background and knowledge.

The grammar is divided in two parts: the lexical grammar and the language grammar. The description in the Listing 3.3 contains the lexical and language grammar rules used to design an interactive experience at a space.

```
%lex

%%
\s+ {/* skip whitespace */}

[0-9]+("."[0-9]+)?\b {return 'NUMBER';}

/* UNIVERSAL POINTS OF INTEREST */
"NO_POI"      {return 'POI';}
"START"       {return 'POI';}

/* TIME */
"WAIT"        {return 'WAIT';}
"DURATION"    {return 'DURATION';}
"NONE"        {return 'NONE';}
"FOREVER"     {return 'FOREVER';}
"MOUSE_CLICK" {return 'MOUSE_CLICK';}

/* BODY GESTURES */
"WAVE"        {return 'BODY_ACTION';}
"RAISE_HAND"  {return 'BODY_ACTION';}
"POKE"        {return 'BODY_ACTION';}
"CLICK"       {return 'BODY_ACTION';}
"TAP"         {return 'BODY_ACTION';}

/* DEVICE GESTURES */
"SHAKE"      {return 'DEVICE_ACTION';}
"TURN_DOWN"  {return 'DEVICE_ACTION';}
"TURN_UP"    {return 'DEVICE_ACTION';}
"POINT"      {return 'POINT';}

/* MOVEMENT ACTIONS */
"STOP"         {return 'MOVEMENT_ACTION';}
"OUT_OF_VIEW"  {return 'MOVEMENT_ACTION';}
"MOVING_AWAY"  {return 'MOVEMENT_ACTION';}
"MOVING_TO"    {return 'MOVEMENT_ACTION';}

/* GENERIC REACTIONS */
"CHANGE_IMAGE"                  {return 'CHANGE_IMAGE';}
"VIBRATE"                       {return 'SHOW_REACTION';}

/* COLLECTING */
"COLLECT"         {return 'COLLECTION';}
"DO_NOT_COLLECT"  {return 'COLLECTION';}
"SHOW_COLLECTION" {return 'SHOW_COLLECTION';}

"->"            {return '->';}
"OK->"          {return 'OK->';}
"NOK->"         {return 'NOK->';}
"WITH"          {return 'NOK->';}
"="             {return '=';}
":"             {return ':';}
"{"             {return '{';}
"}"             {return '}';}
","             {return ',';}
"AND"           {return 'AND';}
"OR"            {return 'OR';}
" WITH "        {return ' WITH ';}
"IFPROJECTOR"   {return 'IFPROJECTOR';}
"IFNOPROJECTOR" {return 'IFNOPROJECTOR';}

/lex

/* operator associations and precedence */
```

```
%nonassoc '{' '}'
%right '='
%left ':'
%left '->'
%left 'AND' 'OR'
%right ','
%left '|'

%start SPACE_EXPLORATION

%% /* language grammar */
SPACE_EXPLORATION : GAME
                  | INTERACTIVE_TOUR
                  | TOUR;

GAME : '{' INTERACTION_SEQUENCE '}' ',' FINISH
     | IFPROJECTOR '{' '{' INTERACTION_SEQUENCE '}' ',' FINISH '}'
       IFNOPROJECTOR '{' '{' INTERACTION_SEQUENCE '}' ',' FINISH '}';

INTERACTIVE_TOUR : '{' INTERACTION_SEQUENCE '}'
                 | IFPROJECTOR '{' '{' INTERACTION_SEQUENCE '}' '}'
                   IFNOPROJECTOR '{' '{' INTERACTION_SEQUENCE '}' '}';

TOUR : '{' POI_SEQUENCE '}'
     | IFPROJECTOR '{' '{' POI_SEQUENCE '}' '}'
       IFNOPROJECTOR '{' '{' POI_SEQUENCE '}' '}';

FINISH : INTERACTION_SEQUENCE ;

VIRTUAL_LOCATION : VIRTUAL_CHARACTER | VIRTUAL_OBJECT ;

POI_SEQUENCE : POI
             | POI_SEQUENCE ',' POI_SEQUENCE ;

INTERACTION_SEQUENCE : POI ':' ACTION TIME '->' REACTION TIME
                     | POI ':' ACTION GAMEPLAY TIME '->' REACTION TIME
                     | POI ':' '{' ACTION_SET '}' TIME '->' REACTION TIME
                     | VIRTUAL_LOCATION ':' ACTION TIME '->' REACTION TIME
                     | VIRTUAL_LOCATION ':' '{' ACTION_SET '}' TIME '->' REACTION TIME
                     | '{' INTERACTION_SEQUENCE ',' INTERACTION_SEQUENCE '}'
                     | '{' INTERACTION_SET ',' INTERACTION_SEQUENCE '}'
                     | INTERACTION_SEQUENCE ',' INTERACTION_SEQUENCE;

INTERACTION_SET : INTERACTION_SEQUENCE 'OR' INTERACTION_SEQUENCE
                | INTERACTION_SEQUENCE 'AND' INTERACTION_SEQUENCE;

TIME : |
     | NONE
     | MOUSE_CLICK
     | FOREVER
     | DURATION '=' NUMBER
     | WAIT '=' NUMBER
     | DURATION '=' NUMBER WAIT '=' NUMBER;

ACTION : BODY_ACTION
       | MOVEMENT_ACTION
       | DEVICE_ACTION;

ACTION_SET : MOVEMENT_ACTION ',' BODY_ACTION
           | MOVEMENT_ACTION ',' DEVICE_ACTION
           | POINT ',' DEVICE_ACTION
           | BODY_ACTION ',' DEVICE_ACTION;

REACTION : SHOW_REACTION
         | SHOW_REACTION COLLECTING
         | QUESTIONING
         | INPUT
         | SHOW_COLLECTION
         | SHOW_REACTION CHANGE_IMAGE
```

```
134         | VIDEO ' WITH ' SHOW_REACTION;
135
136 QUESTIONING : QUESTION
137               'OK->' SHOW_REACTION COLLECTING
138               'NOK->' SHOW_REACTION COLLECTING;
139
140 COLLECTING : | COLLECT | NOT_COLLECT;
```

Listing 3.3: XploreDescription: lexical and language grammar.

Exploring a space can be done in the form of a GAME, INTERACTIVE_TOUR or simple TOUR. A GAME is an application composed by a set of INTERACTION_SEQUENCES, being at least one the FINISH_SEQUENCE. An INTERACTIVE_TOUR is a game without FINISH_SEQUENCEs. A TOUR is a set of POIs to be visited in a given order. All this forms of exploring a space can be defined when visualised using a projector and when visualised using the mobile device display.

INTERACTION_SEQUENCES are trios of points of interest (physical POIs at the space, VIRTUAL_CHARACTERs or VIRTUAL_OBJECTs), ACTION and REACTION, which means that if an ACTION is performed in the surroundings of the point of interest, the REACTION is to be executed by the mobile device. The application runs until the user executes the action of a FINISH_SEQUENCE at the corresponding POI.

ACTIONS can be GESTURE_ACTIONS (for example, wave or poke), DEVICE_ACTION (for example, shake or point) or MOVEMENT_ACTIONS (for example, moving to or away from a POI).

Both ACTIONS and REACTIONS can have a TIME indication, meaning that they will last until, for example a mouse click occurs or a specified duration in seconds ends.

REACTIONS can have a COLLECT indication, meaning that the information regarding the corresponding POI will be collected.

REACTIONS can be multimedia information (including augmented videos), mobile device VIBRATE, CHANGE_IMAGE of a point of interest, questions, requests for INPUT or to show collected items.

A more detailed description of the lexical and language grammar and their usage can be found in Appendix B.

The description in Listing 3.3 up to line 62 contains the basic form of the lexical grammar that is common to all spaces. For each space the grammar has to be created with the correct configuration of Points-Of-Interest, Actions and Reactions. The basic form of the lexical grammar in Listing 3.3 needs to be complemented with the information specific to each space.

Examples of grammars (lexical and language) can be observed in Appendix C. For the applications described in Section 4, the grammar specific information for each space and the application description file are presented.

3.3 Defining a Space and Designing Applications

Applications that are interactive experiences can be described using a Domain Specific Language (DSL) grammar like the one in Section 3.2. Writing the description file of the application based on the grammar is prone to errors and requires the knowledge of how a grammar works. Constructing some rules can be a complex task and we aim at eliminating this difficulty. We want to be able to design interactive experiences in a straightforward way and to be able to generate the corresponding description file without having to write it.

To design interactive experiences and generate their description file, a visual tool, called XploreBuilder, was created. It is a desktop application implemented in Java that provides a graphical user interface to design applications, like games, to make space exploration more interesting and fun. With XploreBuilder anyone with little or no knowledge of programming languages, grammars, mobile device applications or computer networks can design an interactive experience application by assembling building blocks, created based on the initial configuration of the space. XploreBuilder generates an application description file containing the description of the interactive experience application (see Section 3.2) and a space description file (see Section 3.3.1). These files can be passed to another system (for example, urSpace described in Section 3.4) to create the interactive experience applications.

Many applications can be created at a same space differing from each other in the arrangement of the blocks. Existing applications can be updated and new features easily added by re-arranging the blocks or adding more blocks. New building blocks can be created at any given time by providing more information about the space and possible interactions.

An initial research about the space characteristics has to be performed to obtain information to configure the application and define the space (see Section 3.3.1). Since our only requirements is to have a video camera network and a wireless network for communication available, for this configuration the user (besides the content to be transmitted to the visitors) only needs to possess information about the space dimensions, location and specifications of cameras of the video camera network.

This configuration is done once and is the task that takes longer and takes the most effort. After configuration is done, possibilities for applications (combinations of building blocks) are immense (see Section 3.3.2).

3.3.1 Defining a Space

The information about the space and its contents has to be provided to XploreBuilder. This information has to be entered by a user that has a deep knowledge of the space and the information to transmit. Such user is likely to be the space owner, manager or curator. With this information Points Of Interest building blocks can be created. Also, the actions that the user can perform and the possible reactions have to be provided for the corresponding building blocks to be created. This information is also used to generate the lexical grammar for the space (see examples in Appendix C) and two description files. One for the space, in which all that is related to the space is described and another one generated according to the designed interactive experience. Examples of interactive experience description files can be observed in Appendix C.

The space description file is a comma separated value file were every value and entry of the tables shown in Figures 3.6 and 3.7 and information about the space is written in a separate line. Separators with the name of the table or field are placed above the corresponding information. An example can be observed in Figure 3.4. The file's name will be "<space_name>_description.csv" .

```
Space Name
Famous People
Description
UBI, The Guardian Dragon, your virtual sidekick, helps explore the Famous People Exhibition
Number of Rooms
1
Collections
false
Room1
Name
Famous People
Description
Famous Five
Width
5
Height
4
Length
15
Cameras
CAMERA1,0,Webcam,0.0,0.0,1.0,43.0,57.0
CAMERA2,1,Kinect,2.5,0.0,1.0,65.0,80.0
POIs
POI11,Steve Jobs,STEVE_JOBS,,2,15,1.5
POI12,Dom Quixote,DOM_QUIXOTE,,2,0,1.5
POI13,Pythagoras,PYTHAGORAS,,0,13,1.5
POI14,Joan Miro,JOAN_MIRO,,4,0,1.5
POI15,Luis Vaz de Camoes,LUIS_VAZ_DE_CAMOES,,3,10,1.5
Virtual Elements
POI16,UBI,UBI,,Character,0.0,0.0,0.0,100,170,center,0.0,0.0,0.0
Actions
Wave
Poke
Moving Away
Moving To
Stop
Texts
,,,,,,,,
Multimedia
IMAGE101,Image,How many triangles?,How many triangles?,POI103,SPACE2/pythagoras_hint_ballon.png,false,
     false,,
IMAGE102,Image,Hint: look at the map,Hint: look at the map,,SPACE2/finish_hint_ballon.png,false,false
     ,,
IMAGE103,Image,Breath Fire,Breath Fire,,SPACE2/dragon_fire.png,false,false,,
IMAGE104,Image,Wage Tail Down,Wage Tail Down,,SPACE2/dragon_headT_tailD.png,false,false,,
IMAGE105,Image,Wage Tail Up,Wage Tail Up,,SPACE2/dragon_headT_tailU.png,false,false,,
IMAGE106,Image,Flap Wing Down - Tail Down,Flap Wing Down - Tail Down,,SPACE2/dragon_tailD_wingD.png,
     false,false,,
IMAGE107,Image,Flap Wing Up - Tail Down,Flap Wing Up - Tail Down,,SPACE2/dragon_tailD.png,false,false
     ,,
```

```
IMAGE108,Image,Guess the word: 8 letters,Guess the word: 8 letters,,SPACE2/start_ballon.png,false,
     false,,
IMAGE109,Image,Hint: iPod, iPad, iPhone, iMac,Hint: iPod, iPad, iPhone, iMac,POI101,SPACE2/jobs_hint_
     ballon.png,false,false,,
IMAGE110,Image,Correct! You have earned the letter: i,Correct! You have earned the letter: i,POI101,
     SPACE2/letter_I_ballon.png,true,false,TEXT,I
IMAGE111,Image,Second letter of Portugal’s neighbour.,Second letter of Portugal’s neighbour.,POI104,
     SPACE2/miro_hint_ballon.png,false,false,,
IMAGE112,Image,Correct! You have earned the letter: p,Correct! You have earned the letter: p,POI104,
     SPACE2/letter_P_ballon.png,true,false,TEXT,P
IMAGE113,Image,Correct! You have earned the letter: l,Correct! You have earned the letter: l,POI103,
     SPACE2/letter_L_ballon.png,true,false,TEXT,L
IMAGE114,Image,Hint: Look for the name of the farmer on the third paragraph.,Hint: Look for the name
     of the farmer on the third paragraph.,POI102,SPACE2/quixote_hint_ballon.png,false,false,,
IMAGE115,Image,Correct! You have earned the letter: o,Correct! You have earned the letter: o,POI102,
     SPACE2/letter_O_ballon.png,true,false,TEXT,O
IMAGE116,Image,Wrong! Try again!,Wrong! Try again!,,SPACE2/wrong_ballon.png,false,false,,
IMAGE117,Image,THE END! Hope you enjoyed the game. Ask for your prize!,THE END! Hope you enjoyed the
     game. Ask for your prize!,POI105,SPACE2/camoes_ballon.png,false,false,,
IMAGE118,Image,You have all the letters. Find Camoes!,You have all the letters. Find Camoes!,,SPACE2/
     finish_ballon.png,false,false,,
IMAGE119,Image,Map,Map,,SPACE2/map.png,false,false,,
IMAGE120,Image,Background Image,Background Image,Start,SPACE2/background.png,false,false,,
VIDEO101,Video,Video,Video,,,false,false,,
Questions
QUESTION101,What is apple’s most famous letter?,,c,n,u,i,POI101
QUESTION102,What’s the second letter of the author’s nationality in english?,,a,p,a,s,POI104
QUESTION103,How many right-angled triangles are there in Figure 2?,triangles.jpg,a,44,48,40,POI103
QUESTION104,What is the last letter of the first name of Quixote?s companion?,,b,s,o,z,POI102
Requests
,,,,,,,
```

Listing 3.4: XploreBuilder: generated space description file example.

When creating a new space (see Figure 3.5), the user has to provide information about the name of the space, its description and how many rooms it has (the rooms are configured one by one).

When selecting "Create" a new window will open where a tab named content and tabs with the several room names are displayed (see Figures 3.6 and 3.7). One of the main objectives of an interactive experience to help visit a space is to provide additional information to the visitor. Content about the space is very important and has to be provided to the application (see Figure 3.6).

Content can be provided in the form of a text. A text has an unique identification (ID), that is generated by the tool and a title. It can be entered or can be supplied in a file. If the text is related to a Point of Interest or another element of the space, the information about that element (its ID) can be entered.

Other multimedia content can be provided in a file. The types allowed are video, sound and images. Its unique identification (ID) is generated by the application, the title has to be provided and a description may be entered. As in texts, if the content is related to an element of the space, its ID can be entered.

If the applications to be designed will have a collection functionality (see Sections 4.1 and Section 4.3), we can indicate for every Reaction if there is a collectable item associated with it and if it is, if it can be dropped. If there is a collectable item associated then the type of the item (text, image or virtual character) and the item to collect (text or a filename

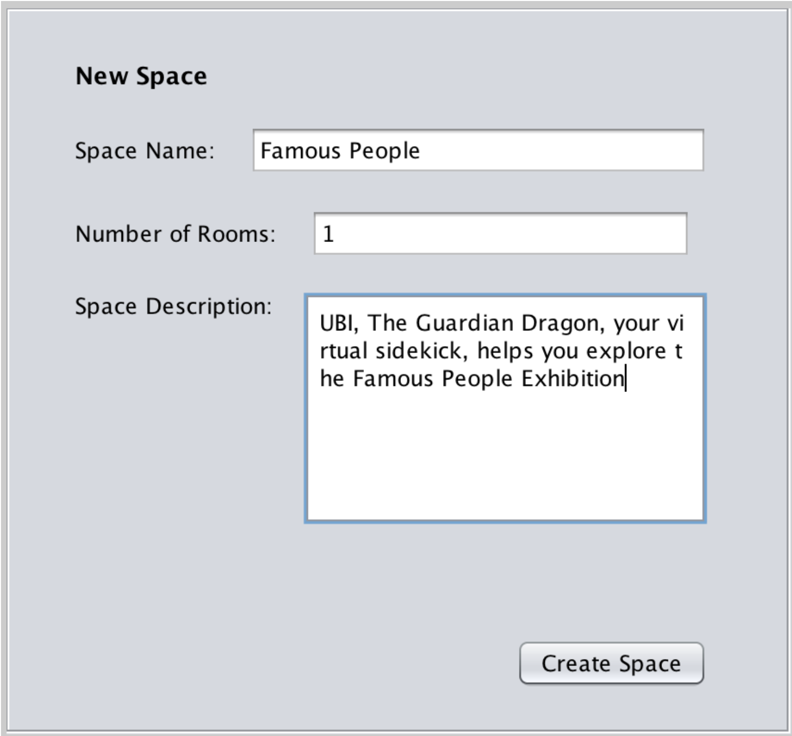


Figure 3.5: XploreBuilder: creating a new space.

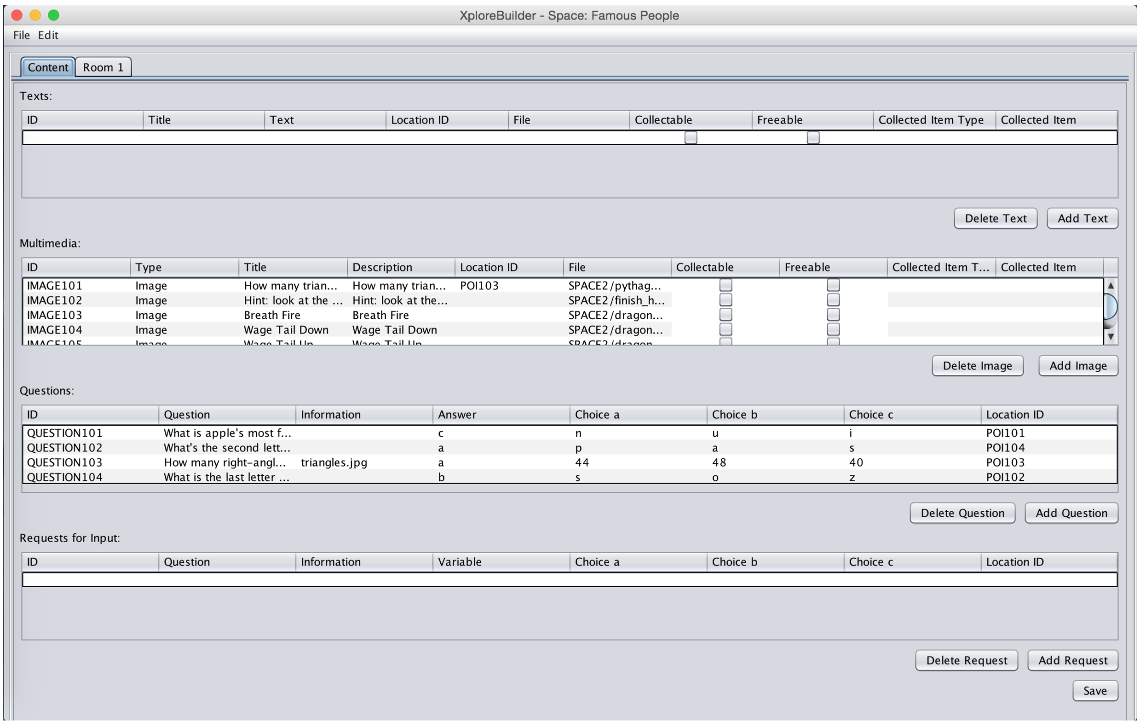


Figure 3.6: XploreBuilder: defining general content.

or the identification of the virtual character) have to be specified. The item remains in the collection until it is dropped, being deleted from the collection. Collecting a virtual element means that it will no longer be available at the space and will become part of the user's collection (examples of this functionality can be observed in the Gone Fishing application in Section 4.3). Once it is dropped it will, again, become part of the space. Since virtual elements can move, the location where it is dropped will be the new location for that element.

As interactive visits to a space often have quizzes, multiple choice questions may be created. Since, the application will run on mobile devices, which have small screens, the question may only have a maximum of three possible answers.

The question and answers have to be entered. A question has an unique identification (ID) generated by the application. Additional information to display, such as an image, can be supplied. The correct answer has to be supplied and it can be "a", "b" or "c", according to the choices (examples of this functionality can be observed in the UBI, The Guardian Dragon application in Section 4.1). If the question is related to a Point of Interest or another element of the space, the information about that element (its ID) can be entered.

Input may be asked (an example of this functionality can be observed in the UBI, The Guardian Dragon application in Section 4.1). The name of the variable to store the entered information has to be supplied. Additional information to display while asking for input may be supplied. The variable has an unique identification (ID) generated by the application.

After providing the content information, the user can save it. The content information is used to generate both the space description file and the application description file. If the application is closed or another option is selected before saving, a dialog will appear asking if the user desires to save it or to discard changes.

Information about physical elements of the space is needed, such as cameras and points of view location, as well as virtual elements to augment the space (see Figure 3.7). This information is to be supplied room by room. Besides the information to transmit to the visitor, the user, only needs to know the information related to the video camera network, such as space dimensions, camera locations and specifications. This makes it possible to adapt the system to any space that has a video camera network installed or, if there is no video camera network present, to adapt the system to a space by installing one. This means that spaces that are not classical visiting spaces (such as museums and art galleries, which are the typical spaces for these kind of applications) like restaurants, schools, gardens, hotels and supermarkets can be used to create interactive experiences to help visit them.

For each room the user has to supply the name and maximum values for the width, length and height of the room. If the room is not a parallelepiped, these values will correspond to the smallest parallelepiped where the room will fit. A description may be supplied.

As one of the requirements of the system is to have available a network of cameras, information about the cameras has to be supplied. Each camera has an unique identification (ID) generated by the application, the address reference for the video stream, the type of

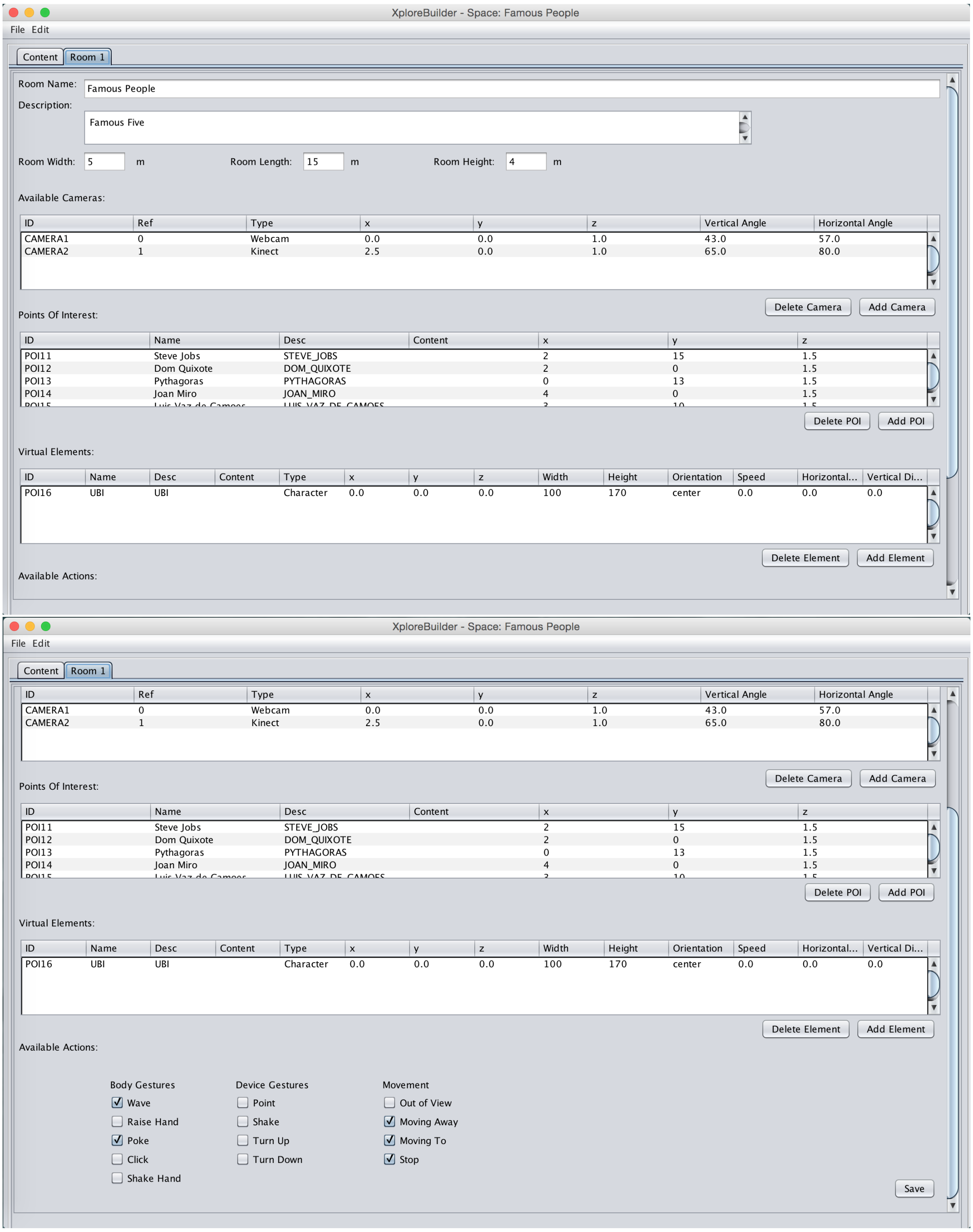


Figure 3.7: XploreBuilder: defining a room.

camera (such as webcam, Internet Protocol or firewire), the (x,y,z) coordinates in the room. The centre of the axis is located at the most southwest, ground level point of the room, being all coordinates positive. The x axis is related to width, the y to length and the z to height. The camera view angle (vertical and horizontal) also has to be supplied.
Points Of Interest (POI) are real locations of major interest at the space, that can not move. A POI can be an object, area of the space (like a vantage point) or anything of major interest. A name and location coordinates in the room (x,y,z) have to be provided. A description may be provided. Every Point Of Interest has a unique identification (ID) generated by the application.
A space can have virtual elements. The type of the element can be character or object. These virtual elements are used to complement the space with content or to provide context to the created application. A name, location coordinates in the room (x,y,z), width, height and orientation of the image have to be provided. If a virtual character is moving right and the orientation of the image is left then the image will be mirrored so the character does not seem to be moving backwards (used in the fish virtual characters in the Gone Fishing application in Section 4.3). A description may be provided. Every virtual element has a unique identification (ID) generated by the application. Virtual elements can move and a direction (composed of two angles where 90° is North for the horizontal angle and 90° is up for the vertical angle) and speed (meters per second) have to be supplied.
The information about cameras and room name, description and dimensions is only used to generate the application description file. The information about Points Of Interest and Virtual Elements is used to generate both description files.
After providing the room information the user can save it. If the application is closed or another option is selected before saving, a dialog will appear asking if the user desires to save it or to discard changes.
The description files are generated when the used choses to create the application in the New Application window described in the next section.

3.3.2 Designing Applications

One of the aims of this research is to provide means to create interactive experience applications by a person with little or no experience of programming. In order for the application to be created, its description files need to be generated according to its design. Designing an application for a space can be done by anyone with access to the Xplore-Builder application and the space configuration. This means that, if it is desired, not only the staff at the space but the public in general can design and create their own interactive experiences to help visit the space. This opens the door to the participation of the public in the visits. An example could be of a teacher of a course preparing a school visit to a space and wanting the students to obtain specific information related to what is being taught.
To design an application the user has to first decide the name and type of application: interactive tour, tour or a game (see Figure 3.8). An interactive tour is a set of interactions

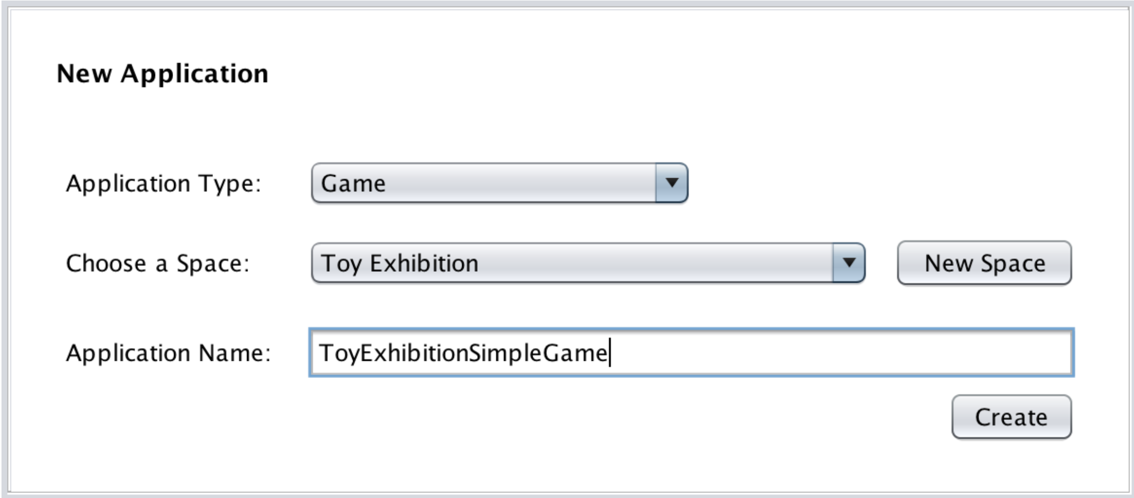


Figure 3.8: XploreBuilder: creating a new application.

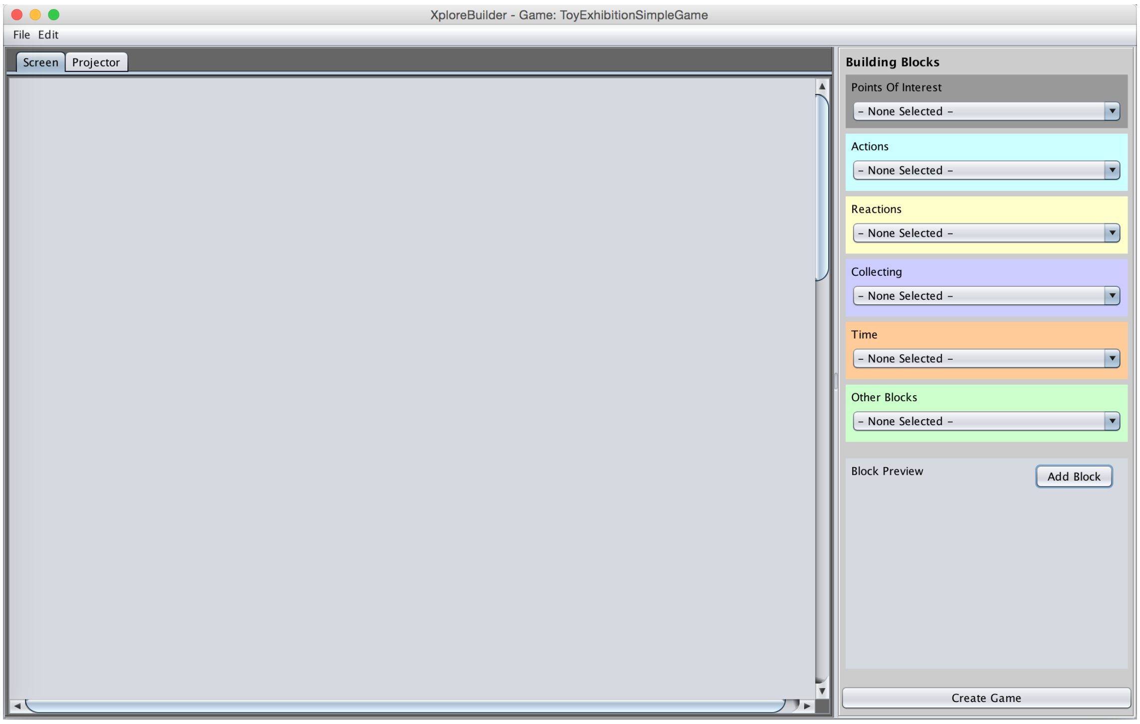


Figure 3.9: XploreBuilder: application design interface.

within a space. Each interaction is called a sequence and is composed of a location (Point Of Interest or virtual element), an action (what the user does to interact with the location) and a reaction (what will happen after the user performs the action). A game is an interactive tour with an additional interaction that defines the end of the game (finish). A tour is a sequence of Points of Interest with no interactions defined.

After filling the form in Figure 3.8, the user is ready to start designing the application (see Figure Figure 3.9). The application is to be executed on a mobile device that may have projection functionalities. The design of the application can be different if designing for a small display or for a projection. To create different designs for the same application, depending on the way the information will be visualised, the tool provides two tabs named "Screen" and "Projector" that will be used as canvas for the design (see Figure Figure 3.9). If one of the tabs is not used, it is assumed that the design will be the same for both.

The interactive experiences are designed by assembling building blocks. The building blocks available are defined according to the initial definition of the space (see Section 3.3.1) and the grammar rules (see Section 3.2). A more complex configuration will make more building blocks become available. This allows the creation of more interesting applications but increases the degree of difficulty of designing.

The space to explore has to have Points Of Interest or virtual elements so interaction can take place. These are represented by grey blocks that will serve as background to the interaction definition.

When an item is selected from the menus on the right, a preview is shown at the bottom of the menus (see Figure 3.10). This is useful to visualise the block before adding it to the design. Some blocks contain large sets of information that would not fit in the menu option (for example, the questions building blocks) and only a part is shown.

To visualise all the information, the item from the menu needs to be selected and the information can be visualised on the preview area. By selecting "Add Block", the block is added to the selected tab on the left.

To describe interactions with the POIs and virtual elements, Action Blocks (see Figure 3.11) are needed as they are the trigger to a reaction of the system, this reaction is represented by a Reaction Block (see Figure 3.12) that will be connected to the respective Action Block.

In case of the reaction having a collectable item associated, a Collect Block can be added to the Reaction (see Figure 3.13) to indicate that the item is to be collected. If nothing is said or if a DO_NOT_COLLECT Block is added then the item is not collected. A Drop Block is also available for when a collected item is to be dropped.

Several actions can be placed in a row meaning that they must all be performed in order for a reaction to take place (see first sequence in Figure 3.14).

A sequence representing an interaction has to have one POI Block and at least one Action Block and one Reaction Block. An application has to have at least one sequence.

If the application is a game, it has to have at least a sequence that determines the end of the application. These sequences contain a Finish Block (see Figure 3.15). By clicking on the desired POI or virtual element block, the sequence will be selected and a Finish Block can be inserted following the same reasoning as for other blocks. It will be placed at the end of the sequence's blocks.

The second sequence in Figure 3.14 could work in a stand alone mode and is an example of the smallest game application that can be created.

If a sequence can only be executed if another sequence is executed first, a Precedence Block needs to be added. The first sequence that needs to be executed has to be selected (see first

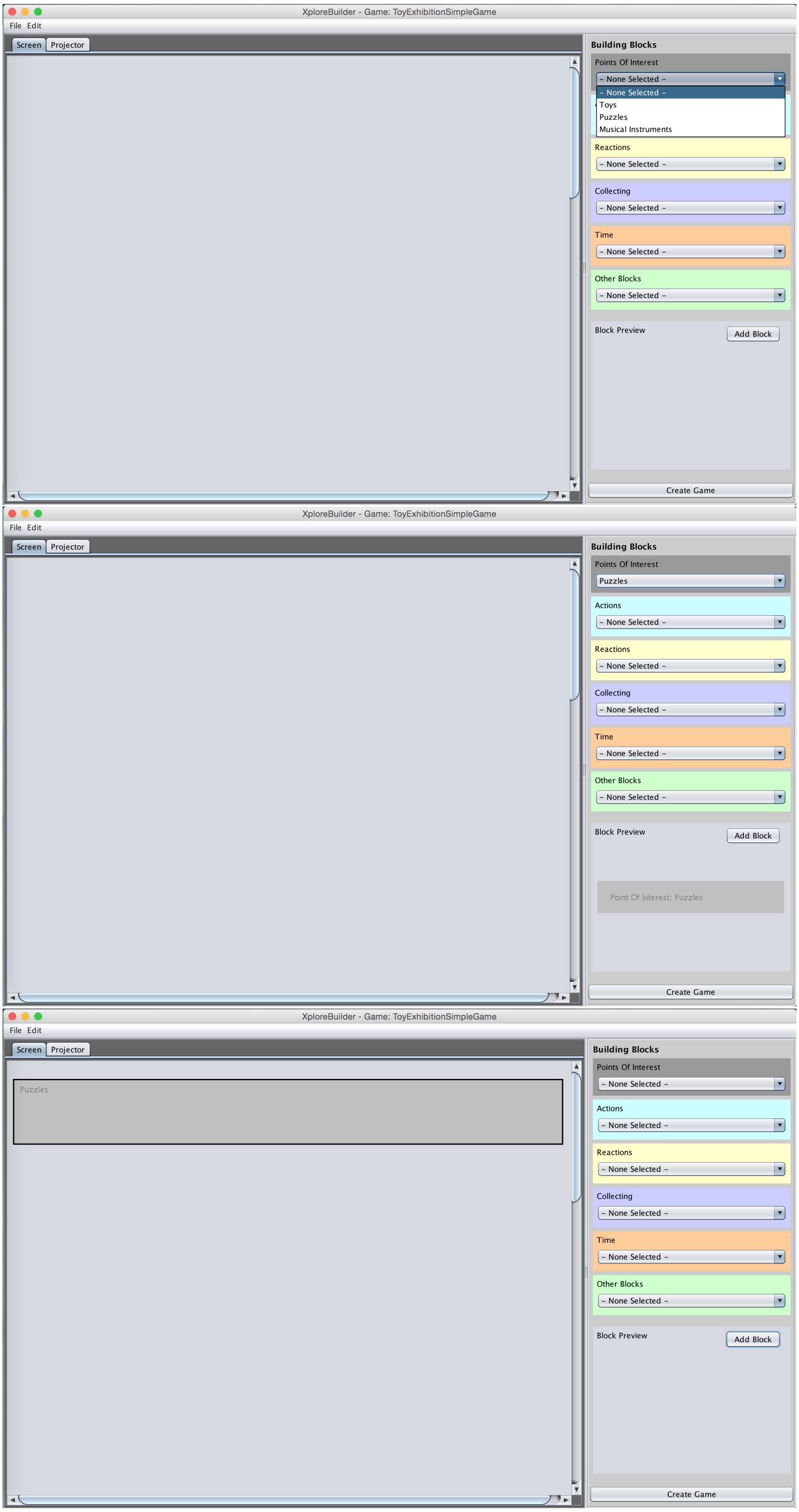


Figure 3.10: XploreBuilder: adding a POI Block.

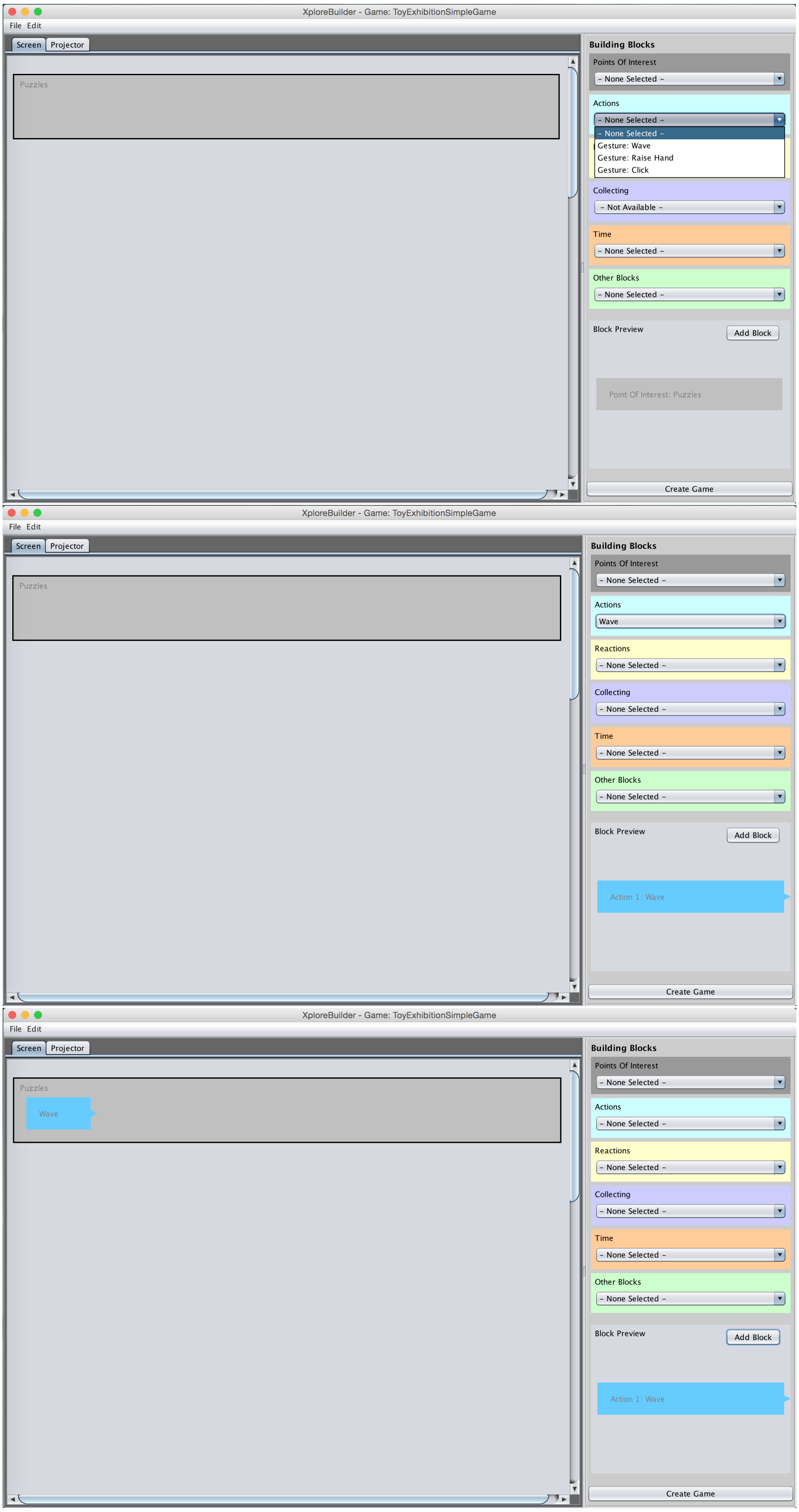


Figure 3.11: XploreBuilder: adding an Action Block.

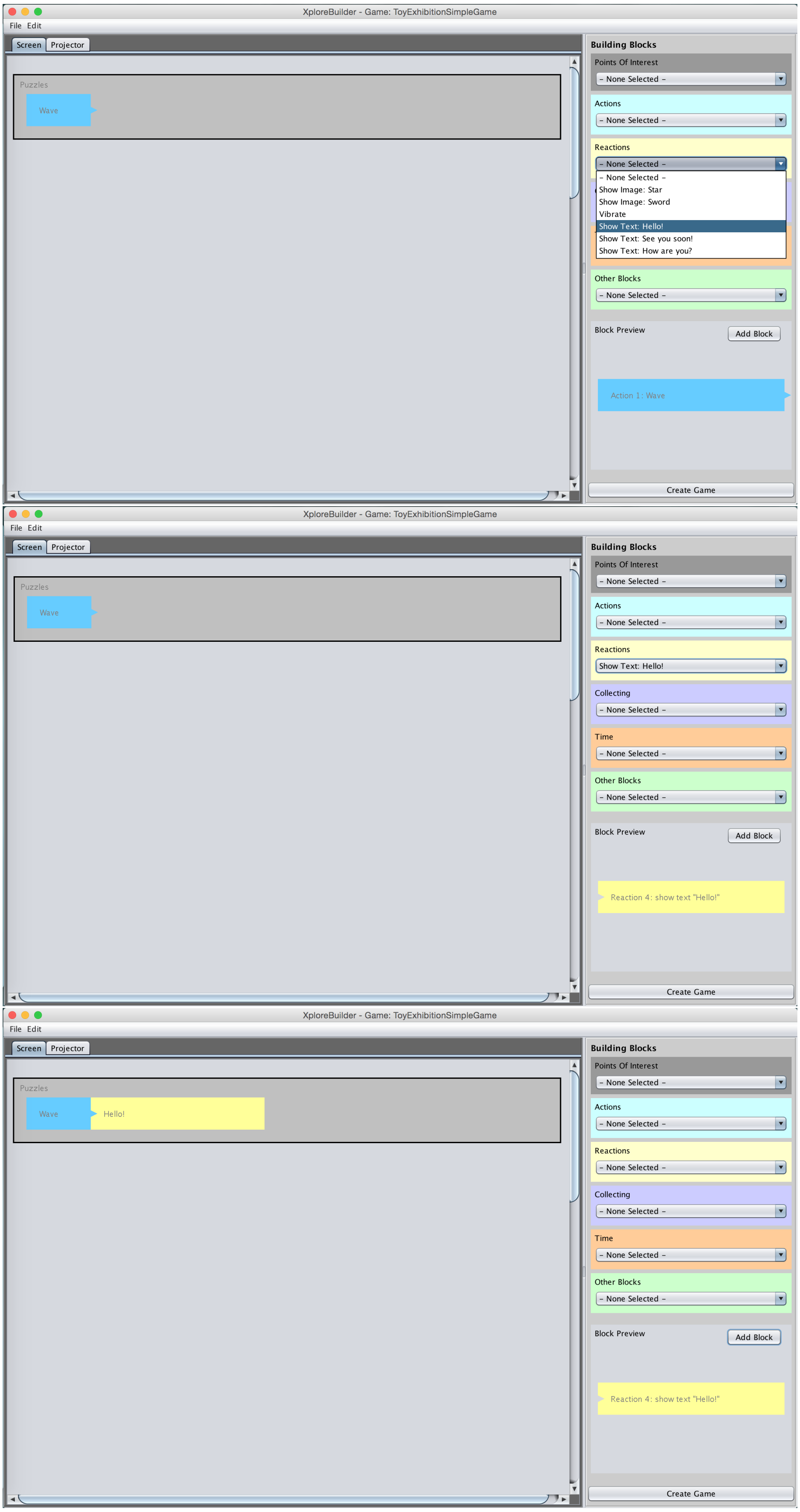


Figure 3.12: XploreBuilder: adding a Reaction Block.

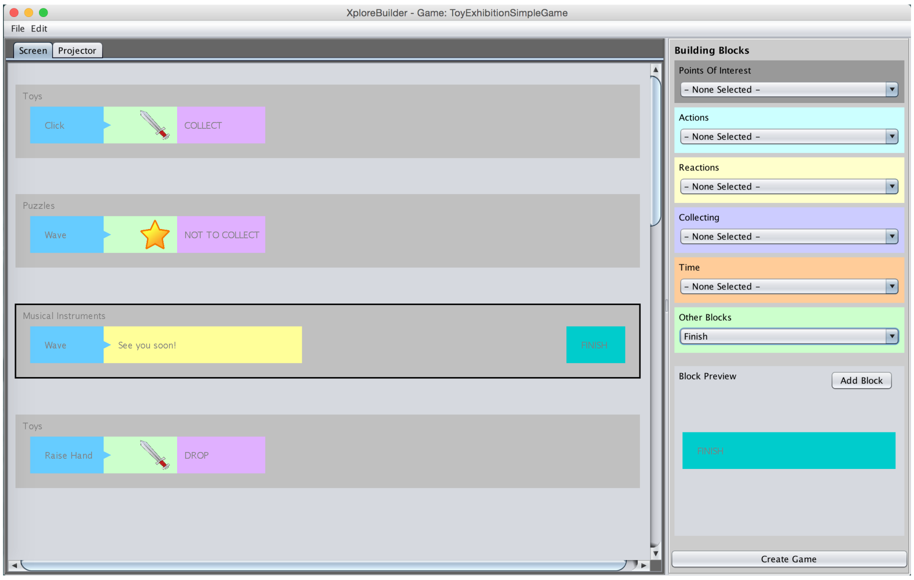


Figure 3.13: XploreBuilder: collections.

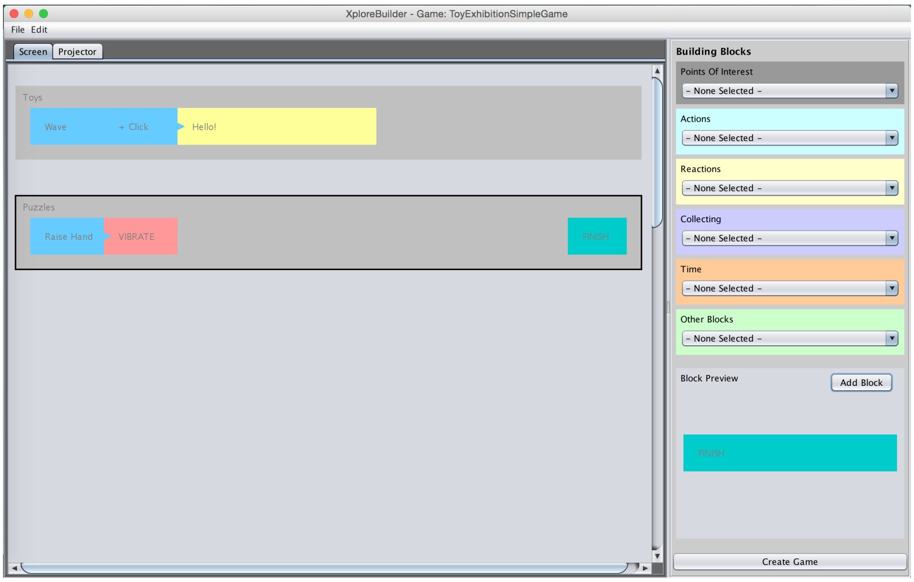


Figure 3.14: XploreBuilder: interaction with two Action blocks.

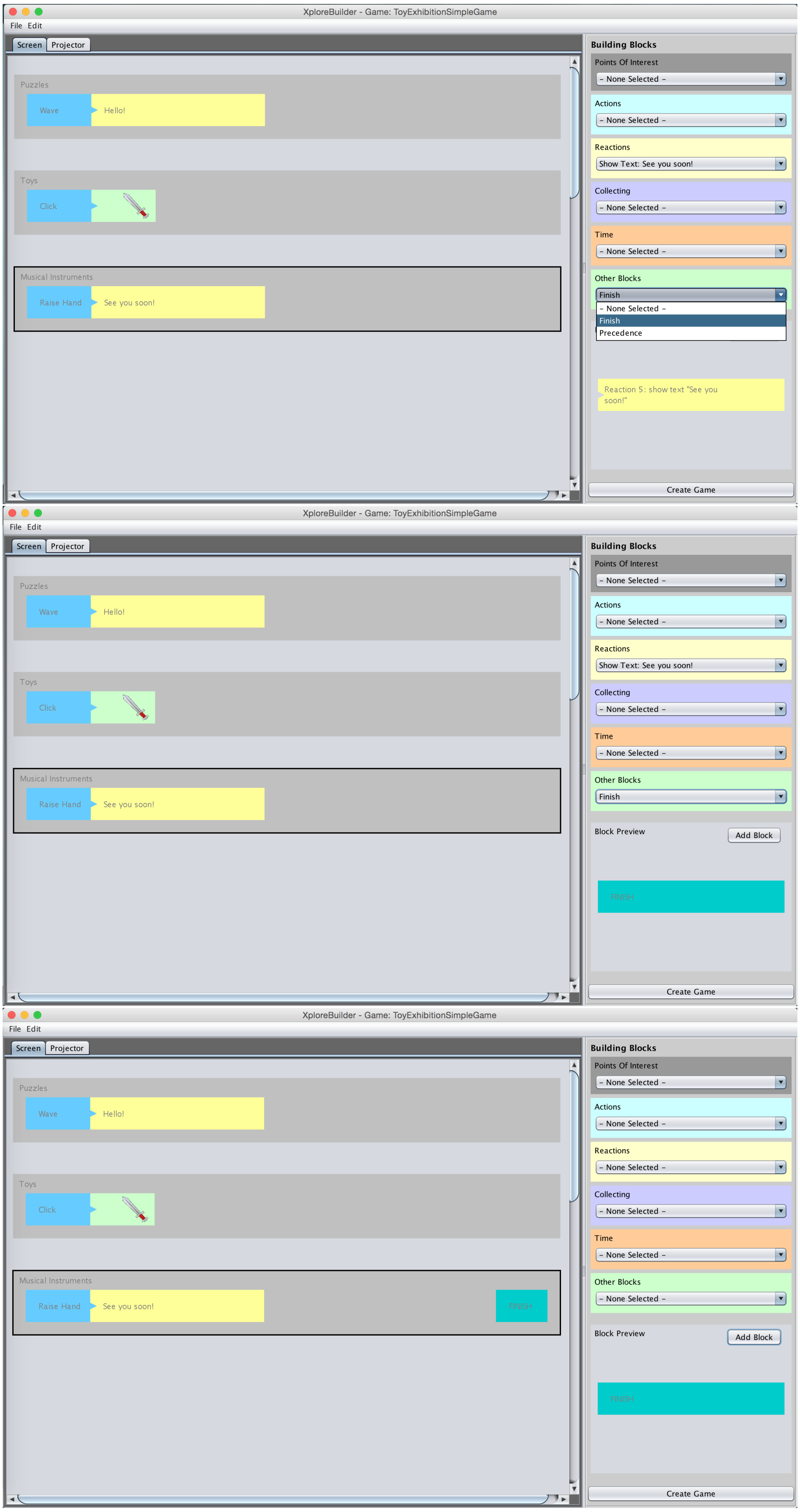


Figure 3.15: XploreBuilder: game finish.

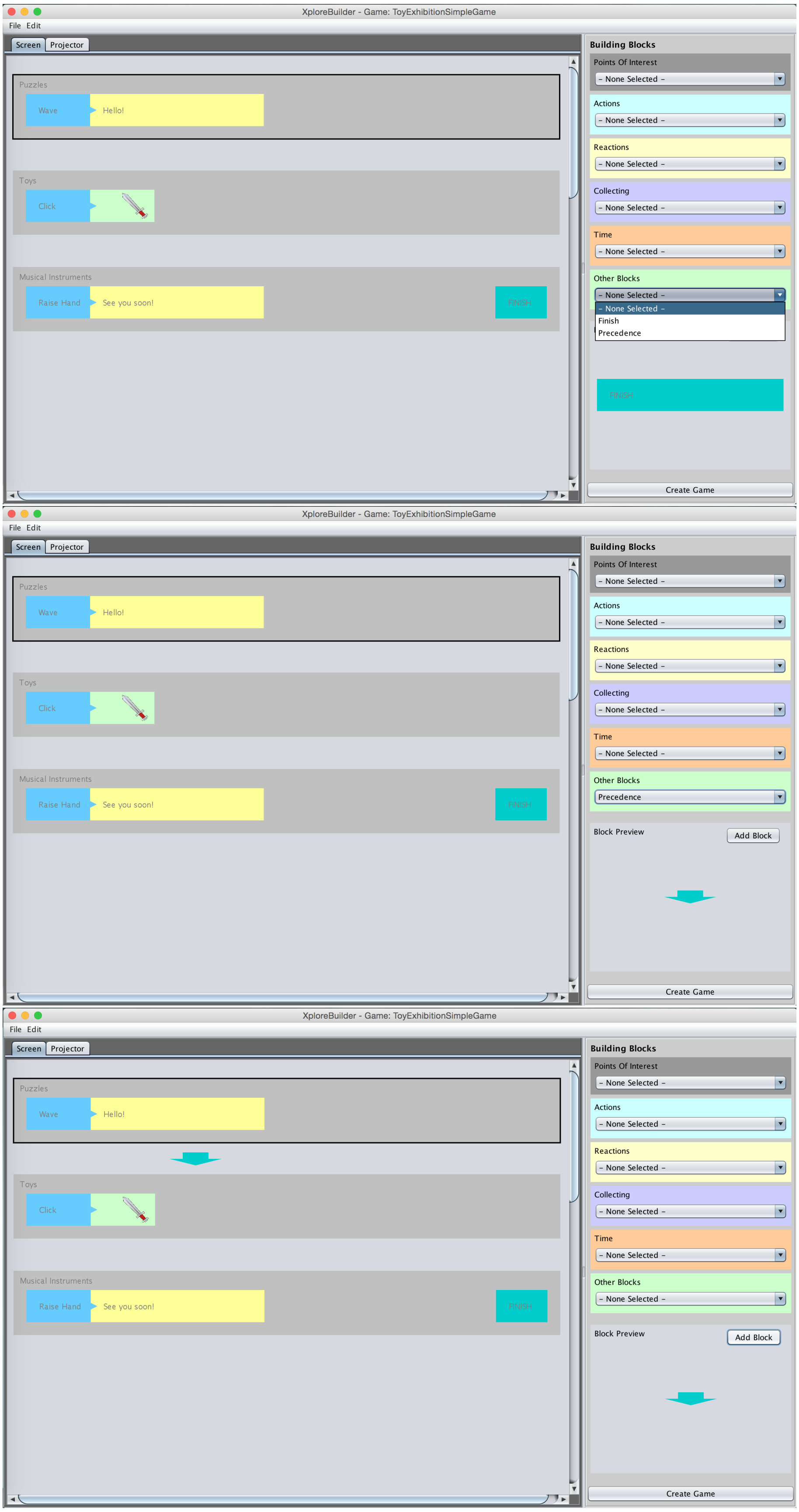


Figure 3.16: XploreBuilder: sequence precedence.

image of Figure 3.16) by clicking on the desired POI or virtual element block.
The Precedence Block is inserted following the same reasoning as for other blocks but it will be located between sequences. There is no limit for the number of sequences with precedences.
When there is the need to define timings for actions and reactions, a Time Block can be used (see Time Block examples in Figure 3.17). Time Blocks define how long an action or reaction take place. There are five Time Blocks. The None Time Block can be used when no timing needs to be defined (it has the same behaviour as not adding a time block to an action or reaction).
The Duration Time Block indicates that the action or reaction will last a given number of seconds. This number is asked to the user.
The Forever Time Block indicates that the action or reaction will last thought the execution time of the application, unless another action or reaction with another Forever Time Block is executed. In this case the second one replaces the first.
The Mouse Click Time Block indicates that the action or reaction will last until a mouse click or tap in case of a touch mobile device.
When two sequences have to be both executed in order to unblock a third one (i.e., to become available), the And Block can be used (see Figure 3.18). Between the two sequences to be executed the And block will be drawn and between the second and the third (the one that will be unblocked) an arrow will be drawn. When to unblock a sequence (make it available) one of two sequences has to be executed, the Or Block can be used (see Figure 3.18). Between the two sequences that can be executed the Or block will be drawn and between the second and the third (the one that will be unblocked) an arrow will be drawn.

The visual development tool complies with the DSL grammar rules (see Section 3.2) for describing applications to minimise any errors writing the application description file.
The generated application description file is a file named "<application_name>_description.grm". Examples of application designs can be observed in Appendix D.

3.4 Creating Applications

The vuSpot infrastructure can be used by people with programming skills (as it was shown when programming the applications in Sections 4.1, 4.2 and 4.3) but one of the aims of this research is to provide means for a person with little or no experience of programming to be able to create interactive experience applications like the ones described.
The XploreBuilder application, together with the XploreDescription, provide means to design and generate descriptions of those applications and respective space they will be used in. This is not enough to create an application. There is the need for another tool that parses the description files and creates the respective interactive experiences, using the

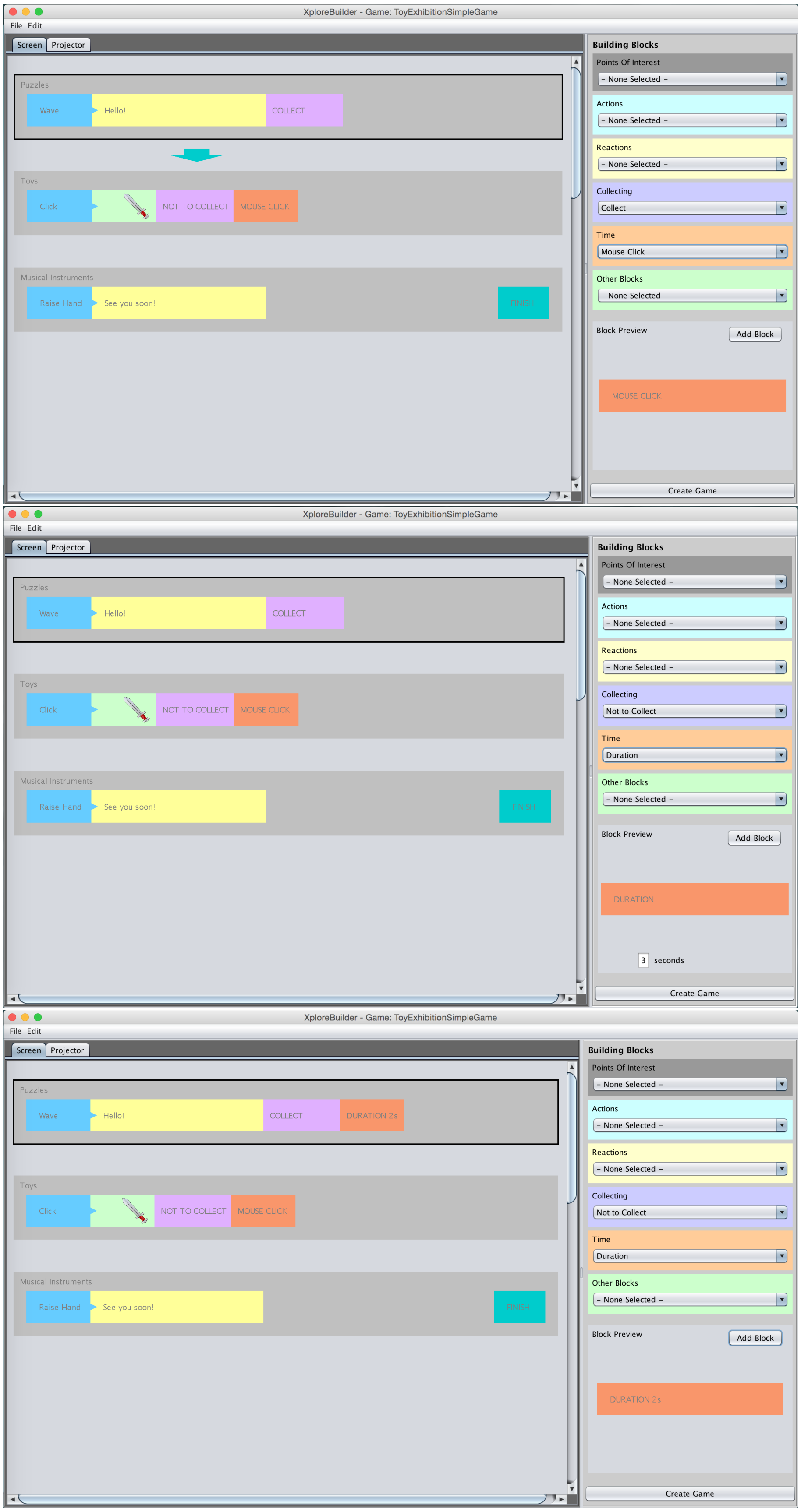


Figure 3.17: XploreBuilder: Mouse Click and Duration Time Blocks.

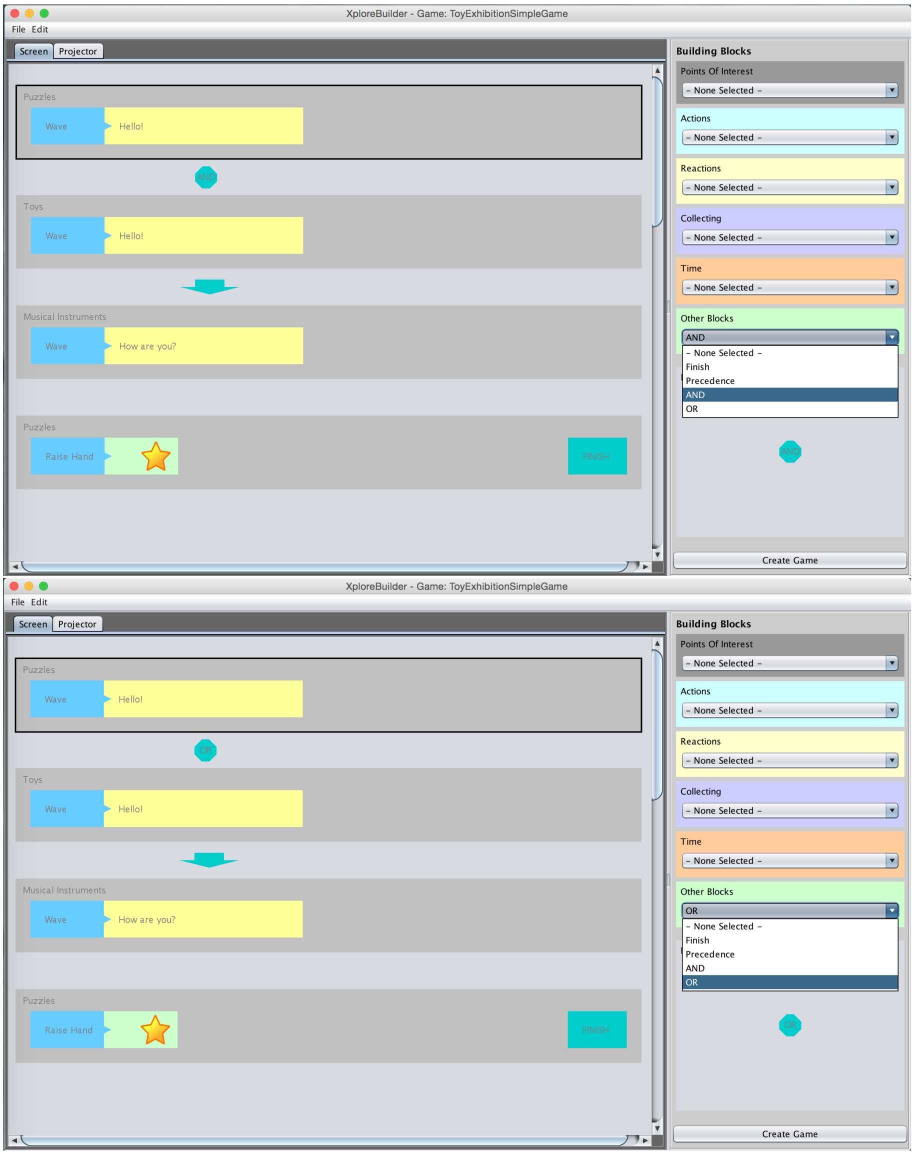


Figure 3.18: XploreBuilder: Top: And Block. Bottom: Or Block.

vuSpot infrastructure to provide means to create all the functionalities described.
urSpace is a tool developed with the goal of simplifying the design and development of applications that use space exploration and interaction with a space as main features.
urSpace uses three elements presented in this research (see Figure 3.19): the vuSpot infrastructure (see Section 3.1), a space description file and an application description file that complies with XploreDescription (see Section 3.2). These files are generated by the XploreBuilder visual tool (see Section 3.3). It can be used by anyone possessing a space description file and an application description file, either if they were written by hand or generated by a tool like XploreBuilder.
It is built on top of vuSpot, which means that it can be adapted to a space provided that the space has a video camera and a wi-fi network installed or can be installed and is

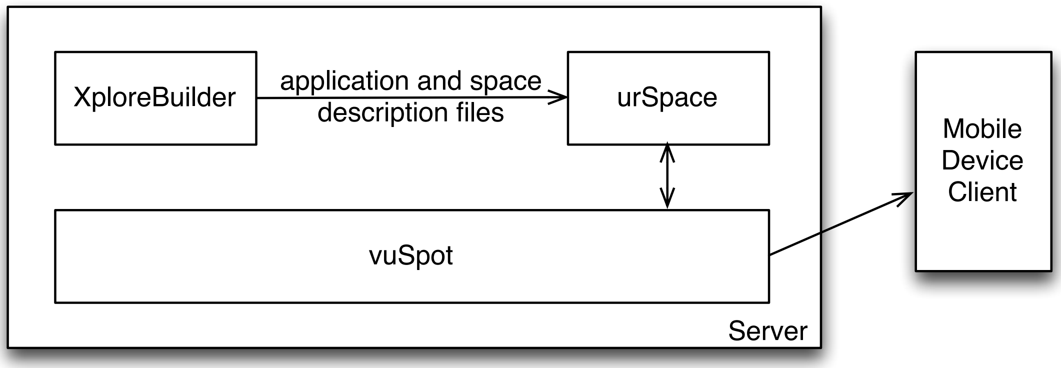


Figure 3.19: System architecture overview.

structured as a client-server architecture.

The XploreBuilder (see Section 3.3) application provides a graphical user interface to design customised interactive experience applications to help explore a space. Two files are generated. A file containing the application description and a file containing the space description. These files are used by urSpace, that parses them and creates the corresponding interactive experience application for the given space, by using the modules of the infrastructure vuSpot. The application created can be immediately accessed using the mobile device client application.

User experiences based on interaction with the space like mobile augmented reality games can be created quickly and without the need to know any programming language. The applications described in Chapter 4 were successfully recreated using the XploreBuilder visual tool and urSpace. The description files and images of the design of the applications can be found in Appendixes C and D respectively.

CHAPTER 4

EXPLORING INTERACTION WHILE VISITING SPACES

In this chapter we present the proof of concept applications. A description of the applications characteristics can be observed in Table 4.1.
All applications were built on top of the vuSpot infrastructure (or part of it) described in Section 3.1. The Toy Exhibition uses the urSpace described in Section 3.4 that is also built on top of vuSpot. They were created to study how interaction can take place and to evaluate if this infrastructure is adequate for the creation of interactive experience applications. Knowledge gained through the creation and tests of these applications was used to adjust and enhance the infrastructure and related tools in Chapter 3. The next section describes UBI, The Guardian Dragon where we intend to explore new forms of interaction using Augmented Reality and mobile devices. MagicLight is a mobile projection system and is described in Section 4.2. With MagicLight we explore the use of pico-projectors and the interaction with a virtual character. In Section 4.3 two applications are described: Gone Fishing and Haunted House. Both these applications use pico projectors to explore a virtual space. In the first case, the user explores the sea bed and in the second a haunted house. With these applications we explore how pico projectors can be used to explore virtual spaces and interact with the virtual elements contained in them. In the last section of this chapter, a Toy Exhibition was created to be used as a space to design interactive experiences using XploreBuilder (see Section 3.3) and create them using urSpace (see Section 3.4).

4.1 UBI, The Guardian Dragon: Your Virtual Sidekick

Many real and fictional characters have sidekicks. Doctor Watson is Sherlock Holmes's sidekick since late 1800 but there is also Don Quixote's Sancho Panza, Batman and Robin among many. Who has not dreamt of having a sidekick like these? Sidekicks help the characters develop the plot of a fictional story and contribute to making it more interesting.

Application	Virtual Elements	Type of Interaction	Social Interaction	Used Tools	Visualisation	User Tested
UBI, The Guardian Dragon	Character	Gestures and movement	Between players	vuSpot	Augmented Reality	YES
Magic Light	Character	Gestures and movement	Between players	vuSpot	Projection	YES
Gone Fishing	Characters Objects	Movement and mobile device sensors	Between players	vuSpot	Projection	YES
Haunted House	Characters Objects	Movement and mobile device sensors	Played in groups	vuSpot	Projection	NO
Toy Exhibition	NA	Gestures and movement	NA	XploreBuilder XploreDescription urSpace	On mobile device Screen	YES

Table 4.1: Description of the implemented applications.

A virtual character can play the role of a user's sidekick. Sidekicks interact with real and fictional characters helping them to overcome challenges. UBI, the Guardian Dragon, is a virtual sidekick that can help a user explore a space.
The virtual sidekick UBI is superimposed on the video stream of the area where the user is located (as stated in Section 3.1 a video camera network and a wireless network have to be present at the space) and this stream is sent to the user's mobile device. The user can see her sidekick by visualising the augmented video stream (see Figure 4.3 (a)) displayed by the mobile device. The information delivered can be adapted to the user age and interests, which must be supplied to the system.
Augmented Reality provides additional information and means to create a virtual environment that meets the user's interests and encourages exploring a space and interacting with it. Interaction can take place with virtual characters, other users and with virtual and real objects of the space, creating a collaborative environment. These characters and objects might introduce historical or science fiction elements and exist to add extra context to the visit. The applications for this system include gaming, virtual tours, entertainment and education.
The main questions addressed in UBI, The Guardian Dragon [SC12] are:

- Exploring and developing alternative solutions for interacting with virtual characters.

- Evaluate the behaviour of the infrastructure vuSpot when developing mobile augmented reality applications where interaction with virtual characters and objects is present

- Evaluate if the infrastructure provides means for interaction with virtual characters and objects.

4.1.1 Application Description

A mobile device application, in this case a game, was created to help explore a simulated art gallery showing an exhibition about famous people. In this gallery there are five Points Of Interest (POIs) scattered throughout the space for the participant to visit: Joan Miró, Don Quixote, Pythagoras, Steve Jobs and Luís Vaz de Camões. Each POI is physically represented by images and text on a wall or object.
We created a mobile device application called "UBI, the Guardian Dragon". This application allows the user to play a puzzle game called "Guess the Word". In the game, the participant has to guess the eight-letter word "lollipop".
Each Point of Interest (POI) had a quiz related to that POI for the participant to solve (see Figure 4.1). If the participant answers the enigma correctly (using the touch functionality of the mobile device to choose her answer from a multiple answer list), she earns a letter. The goal of the game was to guess or earn all the letters of the word. UBI's role in the

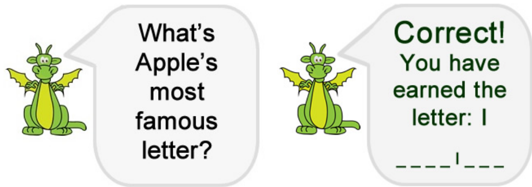


Figure 4.1: Visualization of an enigma and reaction to the correct answer.

Participant's action	UBI's reaction
wave	shows clue
poke/slap	breaths fire
wrong way/wrong action	breaths fire
correct way/correct action	wags tail

Table 4.2: Participant's actions and correspondent UBI's reactions.

game is to show the enigmas, tell the participant if she has got the answer right, give hints if she requests it and guide her through the exhibition.
When the participant approaches a POI, UBI shows an enigma, which consists of text written on the display of the mobile device (see Figure 4.1). POIs can be visited in any order. All POIs except Luís Vaz de Camões POI have a puzzle each, one for each distinct letter of the word. The Luís Vaz de Camões POI is the location of the end of the game, so it does not have an enigma.
When a puzzled is solved, UBI shows a letter. If the participant does do not know the answer and needs a hint, she can ask the dragon by waving at the location he appears to be at the space.
The participant has to estimate the location of UBI in the real world by visualising UBI in the video stream, being shown in the mobile device, and imagine the virtual character at that location in order to be able to perform the gesture correctly (see Figure 4.4).
If the user looks at the display of the mobile device and sees UBI to her left, she can wave to her left at UBI, and he will say "Hello!" and flap his wings. If the user starts moving, UBI can suggest the best route to complete the challenge. When a user stops near a point of interest, UBI gives more information about it. UBI does not like to be poked (or bumped into) (see third image of Figure 4.4). When this happens he gets angry and breaths fire.
Multiple choice values for the answer are shown at the display of the mobile device and one has to be selected using the touch screen. When the participant discovers the four letters she heads to the final location (POI Luís Vaz de Camões) to get a prize.
UBI helps the participant to go to the end (see Table 4.2) by wagging his tail if she is going on the right direction or breathing fire if not. When UBI wants the attention of the participant he flaps a wing. If at any point the participant does something that UBI does not like he breaths fire and if the user does something he likes (like pet him) he wags his tail.

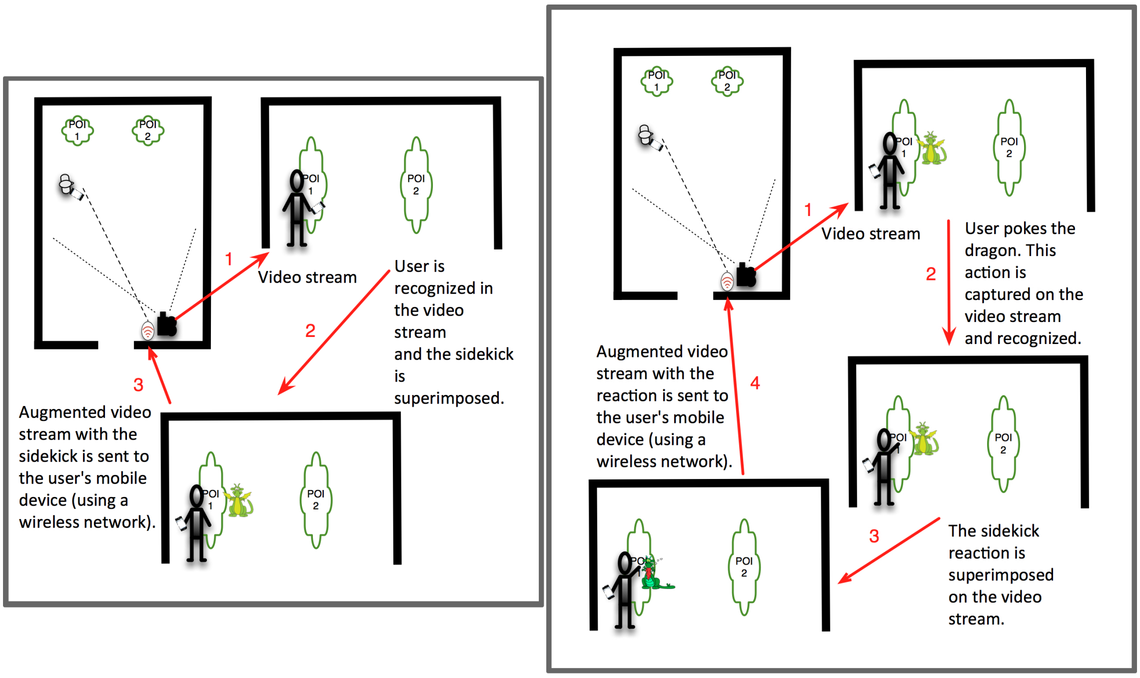
Figure 4.2: UBI, the Guardian Dragon.

4.1.2 Implementation

The hardware used was an iPhone 4S smartphone, a 13” MacBook Pro laptop, a Microsoft Kinect [Micb] camera model 1414 and a firewire video camera. When in the outdoors the information from the user mobile device’s GPS can be used along with blob detection and tracking. When no GPS information is available tracking of the user is made by inferring the user position from the video streams using face recognition algorithms and detection and tracking of blobs. This approach is prone to errors. These errors were overcome with the use of Microsoft Kinect cameras in order to get a notion of depth.
An ad-hoc wireless network was created for the mobile device and the laptop to communicate with each other.
The application was created without the use of the virtual tool proposed in Section 3.3, which means that it was created by a person with programming experience. The results of the tests were used to create the virtual tool and the other applications proposed in Chapter 3.
A version of the game created for the user tests was later designed and created using the proposed applications to verify their behaviour. The description files generated by the XploreBuilder application can be observed in Appendix C and the design in Appendix D. The interactions with UBI were successfully designed and created.

4.1.2.1 The Sidekick Character

A sidekick is a character that is attractive, charismatic and reliable. The user needs to feel comfortable with the sidekick and it should induce a feeling of trust. We created a guardian dragon, called UBI (Figure 4.2).
The dragon was chosen because of the image of strength and wisdom it possesses. In the interviews conducted, 69% all of the interviewees responded positively to the use of a dragon (see Figure 4.12). The interviewees that responded negatively preferred an animal image and none preferred a humanoid character. In the interviews, the qualities most frequently chosen for the sidekick were: happy, nice, colourful and discrete.
In fiction writing [Con03] there are several types of dragons. A Guardian Dragon is considered a playful and curious dragon that likes to observe humans and enjoys their company. They are faithful protectors and have powers that grow as they interact with



(a) Superimposing the sidekick.

(b) Superimposing the sidekick's reaction to the user's action.

Figure 4.3: Interaction.

humans. They are invisible and quiet, except for some moments in which they can be very exuberant making noises and becoming visible to some people. They have a good sense of humour and love pranks. Because of these characteristics the guardian dragon was chosen for the sidekick character.

Dragon characters tend to show some aggressiveness, which is not a good characteristic when we want the user to interact with the character. We chose to make it look childish since it is supposed to inspire happiness and not fear and yet maintain the dragon personality.

Dragons come in all kind of colours. The colour green was chosen because it is a colour associated with calmness and nature (and hope). The sidekick is not a user avatar so it does not have to follow avatars' design guidelines although some of them can be used as hints to what facilitates interaction [Bob+08; Wag+06]. As a first approach for the animation we chose a 2D graphics of the dragon. It simplifies the computation and is able to provide the user the necessary information for the tasks.

4.1.2.2 Interaction

The visitor can see her sidekick by visualising the augmented video stream (see Figure 4.3 (a)) displayed by the mobile device. These characters might introduce historical or science fiction elements and exist to add extra context to the visit. By performing actions in the surroundings of the sidekick or with the use of the mobile device input capabilities, a visitor can interact with it. The system recognises those actions, by processing the video

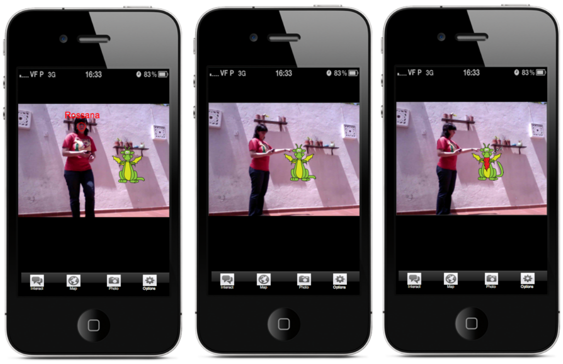


Figure 4.4: User recognition and superimposing the sidekick. Example of interaction: poking.

stream, and produces a reaction of the sidekick that is superimposed on the video stream and can be visualised on the mobile device (see Figure 4.3 (b)).
A guided tour can be taken by following the instructions of the sidekick and receiving from it additional information about the space and points of interest. Taking such a guided tour has advantages like avoiding crowded points of interest.
Other information can be superimposed onto the video stream. Additional virtual characters can be found along the tour to give emphasis to the context, for example, how people dressed or how an handcraft was done. A more interactive way of visiting the space is playing a game. The sidekick plays an important role while playing a game. Beside the virtual information available for all visitors, the user can get more information through the sidekick that can give hints that help the visitor to discover hidden details of the space and guide the visitor in the quest for answers to quizzes.

4.1.2.3 Scenario of use

It is possible to interact with a virtual character, called UBI (Figure 4.2). UBI is a Guardian Dragon. When a user is recognised in the video stream, UBI is superimposed (see first image of Figure 4.4) and interaction is recognised making UBI react.
To use the system, a visitor, has to be identified through a pre-registered username. The information about the user interests has to be also provided. When looking at the display of the mobile device, a video stream of the location where the user is will be visualised. This video stream is captured by the video camera network present at the space and augmented with information relevant to the user. In this video stream the user will also visualise her sidekick (see second image of Figure 4.4).
Interaction can be done by performing movements within the location where the (invisible) sidekick is in the space. Using algorithms to recognise user actions (for example, tapping or waving hello in the surroundings of the sidekick) in the video stream [Wre+97], a reaction to those actions can take place (see Figure 4.5).
If the user looks at the display of the mobile device and sees UBI to her left, she can wave at UBI, and he will say "Hello!" and flap his wings. If the user starts moving, UBI can

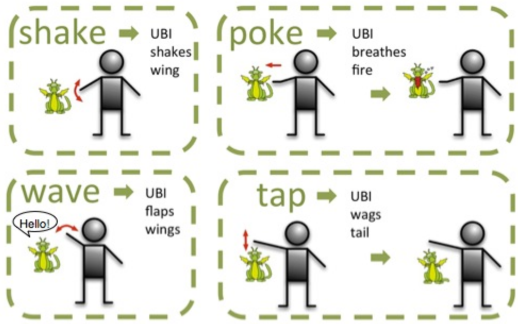


Figure 4.5: Examples of interaction with the virtual character UBI, the Guardian Dragon.

suggest the best route to avoid crowded places. When a user stops near a point of interest, UBI gives more information about it. UBI doesn't like to be poked (or bumped into) (see third image of Figure 4.4). When this happens he gets angry and breaths fire.
The UBI's reactions can be seen in the mobile device display also superimposed onto the video stream. Other virtual elements can exist and enhance the visiting experience providing extra information to the visitor. They can give information about themselves or about the space.

4.1.3 Evaluation

In this section, a summary of the user test results will be presented. The goal of this user test is to evaluate how users interact with the virtual character in the proposed video based environment and to evaluate if the infrastructure modules were adequate for building augmented reality applications.
We also intended to evaluate if the gestures the participant has to make are adequate and comfortable and if interacting with an "invisible" character present at the space would be a good experience. The participants were also questioned about characteristics of the virtual character.

4.1.3.1 Participants

UBI, the Guardian Dragon was tested with thirteen participants. The participants were three females and ten males with ages ranging from 24 to 42 years old and with an average age of 32,2 years. All the participants use mobile phones regularly and eight own smart phones.

4.1.3.2 Test Setup

The setup of the test (see Figure 4.6) simulates the gallery showing an exhibition about famous people described in Section 4.1.1. Participants were asked play the word guessing/quiz game. After playing the game, the participants answered a paper questionnaire

Figure 4.6: Test setup.

regarding the experience of interacting with a virtual character "UBI, The Guardian Dragon" present in the game.
The test was conducted following a "Wizard of Oz" approach. The "Wizard of Oz" technique enables unimplemented technology to be evaluated by using a human to simulate the response of a system. This decision does not influence or alter the information we wanted to gather from the test.
The reason for the user test to follow the "Wizard of Oz" methodology was because gesture recognition of the user and superimposing reactions of the dragon on the video stream were taking too long and this delay was compromising the usability of the application. To ensure a stable behaviour of the dragon, part of the architecture modules were not used. Instead the reactions of the dragon were triggered while observing the participant and immediately sent to the mobile device.
This issue was solved after the test by changing the server to a better computer and changing the communication protocol and using compression algorithms for the video frames transmitted to the mobile device. The test was not repeated because this approach does not influence or alter the information we wanted to gather regarding user interaction. The part of the modules that were not used, were tested afterwards with this and other applications and behaved correctly (these modules are fully tested in the applications described in Sections 4.2 and 4.4).

4.1.3.3 Results

In this section we will discuss the results of the user test. All users did not have problems understand UBI's reactions and all reached the end of the game successfully guessing the hidden word. Hints were obtained from UBI by waving at him. Since waving is a common human gesture we wanted to assess if waving would be an adequate gesture for obtaining information from the virtual character.
Only 8% of the participants considered that waving for hints was difficult (see Figure 4.7). 69% considered it easy or very easy and 23% had an average difficulty. We considered that waving can be used for interaction although participants are not comfortable being seen waving at an invisible being.
Not all participants (31%) tried to make UBI angry (see Figure 4.8). The ones that tried did

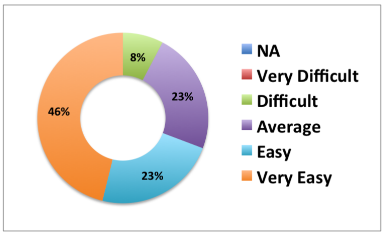


Figure 4.7: Question: Is it easy to obtain a hint from UBI?

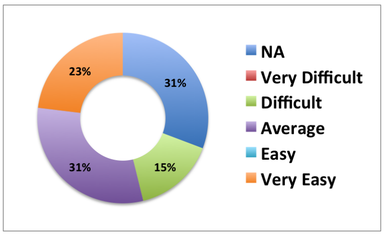


Figure 4.8: Question: Is is easy to make UBI angry?

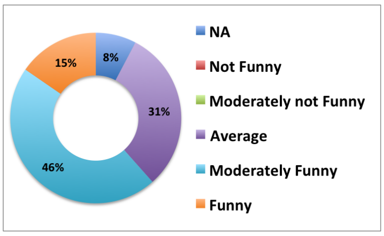


Figure 4.9: Question: Is is fun to interact with UBI?

it by poking or slapping him. We noticed that one of the first reactions of some participants was to try to make him angry. We concluded that a reaction to violence has to be always implemented to indicate to the participants that they are not being nice to the sidekick. 23% considered easy to make UBI angry, 31% had an average difficulty and 15% considered it difficult.

61% of the participants had fun interacting with UBI (see Figure 4.9). 31% considered it average. Some of these participants did not use all means of interaction with UBI available (see Figure 4.8) and considered that it would have been funnier if they did.

UBI also interacts with the participant. He accomplishes this by breathing fire if the participant is doing something wrong or wagging his tail if the participant is performing

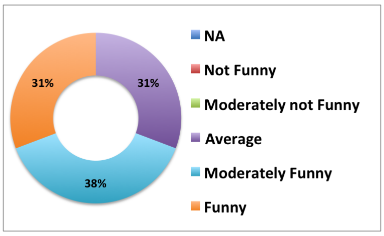


Figure 4.10: Question: Is it fun UBI interacting with the user?

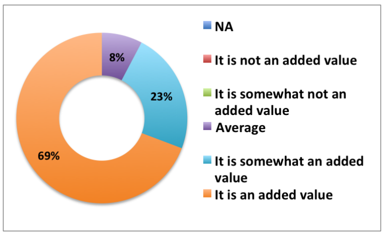


Figure 4.11: Question: Is the interaction with UBI an added value to the game?

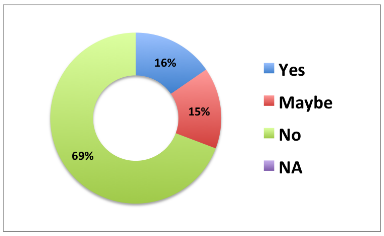


Figure 4.12: Question: The virtual Character of the game is a dragon, an imaginary creature. Would you prefer another kind of image?

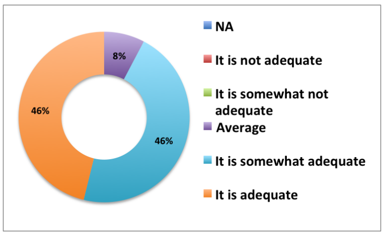


Figure 4.13: Question: Is the communication of the enigmas and hints (using text) adequate to the context of the game?

a task correctly. 31% of the participants considered it funny, 38% moderately funny and 31% average (see Figure 4.10). With these results we concluded that interaction with the participant is very important and a big part of the attractiveness of interacting with a virtual character comes from the expectation of it interacting unexpectedly at some point in time.

The results show that the large majority of the participants prefer to play the game while interacting with a virtual character (see Figure 4.11).

Another question asked was "Would you recommend an application with interaction with virtual characters to your friends?". 77% of the participants would recommend it and 23% might recommend it. Interaction with virtual characters can be an added value to a game played in a space and participants had a positive reaction towards the concept.

We wanted to know if the participants would prefer that the character were a different image than a dragon. 69% of the participants consider that a dragon image works well for the purpose (see Figure 4.12). 15% might prefer a different image and 16% prefer another image. These last participants prefer the image to be of an animal.

The communication of the enigmas and hints was done using text that could be read from the display of the mobile device (see Figure 4.1). Other possibilities like audio

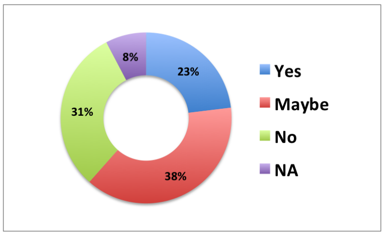


Figure 4.14: Question: Are you willing to carry more devices with different sensors if it would give more functionalities to the application?

could be used. The great majority of the participants consider it adequate or somewhat adequate (see Figure 4.13). The comments they gave were that if audio was used it would be difficult to remember the enigmas and hints but it would be helpful to hear them. Another disadvantage would be that the hints could not be images. A mix of both forms of communication appears to be the best solution.

Participants were asked if they were willing to carry more devices with different sensors if it would give more functionalities to the application. In this question, opinions divide, 31% of the participants are not willing to carry more devices and 23% would (see Figure 4.14). 28% answered maybe. The most common reason for the rejection or doubt was the size and weigh. If the extra devices could be small and light more users would be willing to use them.

92% of the users considered that the vibration functionality would be a major added value to the interaction and 8% are in doubt. When visiting a space the attention should be focused on the space and points of interest of the space and not on the mobile device. Without a way of getting the attention of the user to the virtual character, participants tend to keep on looking at the display to check if the character is interacting. This action is distracting and has to be avoided.

As for the infrastructure, there were some issues concerning the delay of video streaming to the mobile device. The delay in the video was influencing the sidekick's reaction time and therefore the user's experience was being compromised. This was the main reason for the user test to follow the "Wizard of Oz" methodology. We were using the Transmission Control Protocol (TCP) to stream the video frames and an outdated computer. In the months following the user tests most of the issues were solved by compressing the frame data, using the User Datagram Protocol (UDP) instead and using an up to date computer. The rest of the infrastructure behaved correctly.

4.1.4 Results and Discussion

The computational capacity of the computers linked to a video camera network system is used to support the computations needed for the information augmentation that cannot be done by mobile devices. This allows the creation of more interesting mobile applications with more functionalities and make demanding computations faster.

With this game we have shown that augmented reality allows the creation of virtual

characters, like UBI, providing additional information and means to create a virtual environment that meets the user interests and encourages exploring the space. This virtual environment allows the interaction with the characters present in it.
To make a visit of a space more interactive, games can be played and virtual characters and objects can give hints to guide the visitor to another location in the space and point out interesting details of the space that could not be noticed easily.
We evaluated how users perceive interacting with virtual characters. The results showed that the participants are fond of interacting with virtual characters. The words: innovative, fun, easy to use, pleasant and simple were used to describe the experience of using the application to play a game and interact with a virtual character. They also thought that a dragon is an adequate image for a virtual character such as a sidekick.
The tested modules of the infrastructure (see Appendix A) had a good response and the results are encouraging. The infrastructure proved adequate for the creation of augmented reality applications and allowed the successful interaction with virtual characters.

4.2 MagicLight, Bringing Dæmons Into Real Life

MagicLight is another example of an application that can be built using the vuSpot infrastructure (see Section 3.1). The main research questions addressed in this application are, first, how to interact with virtual characters using augmented reality, mobile devices and pico projectors, secondly, evaluate the response of the infrastructure for creating such applications and, thirdly, evaluate if the infrastructure provides means for interaction with virtual characters and objects. MagicLight can be used for gaming, to give additional information about a space and guided tours.
The idea of a presence that guides, motivates and inspires is ancient. The first references to these presences called dæmons come from the greek philosopher Socrates who claims to have an inner voice to guide him throughout every decision nearly 2000 years ago. Plato writes that a dæmon is an autonomous spirit that receives a human at his birth and who follows in life. Leonardo da Vinci's "Lady with an Ermine" (1489-90) and Hans Holbein the Younger's "Lady with a Squirrel and a Starling" (1526-8) inspired Phillip Pullman when creating the "His Dark Materials" novels (see Figure 4.15). "The Golden Compass" movie was based on the first novel of the trilogy, called "The Northern Lights" where every character possess an animal, which Pullman called a dæmon, with its own personality, whose mind is connected to the owners mind and experiences the owner's feelings.
Another series of fantasy novels that inspired movies and whose characters possess a pet companion that helps the owner by performing tasks is J. K. Rowling's Harry Potter.
Virtual pets can be pet toys like Furby [Has] or Tamagotchi [Banb]. Can be played with consoles like Nintendos' Nintendogs + cats [Nin] or Microsoft Xbox's Kinectimals [Mica], online like Zynga's [Zynb] old social network game PetVille or as video games like FishCo [Gam], Pokemon [Com] and The Sims: Pets [Art].
Another example of dæmons are the Digimons [Bana]. Digimons are virtual pets (inspired

Figure 4.15: (a) Leonardo da Vinci "Lady with an Ermine" (1489-90) (b) Hans Holbein the Younger "Lady with a Squirrel and a Starling" (1526-8) (c) His Dark Materials (d) Harry Potter

by the Tamagotchi toys) raised by humans called "Digidestined" or "Tamers". They are fighting dæmons that work together to defeat villains.

It would be amazing if we could have a dæmon in real life to be a companion and help with some tasks, like Harry Potter has, and that can be an extension of one's personalty like the Pullman's dæmon. With this thought in mind we created MagicLight. With MagicLight a new kind of dæmon is created. One that can only be seen through the light of a projector. As benevolent or benign nature spirits, these dæmons take the role of spirit guides to help their owners. The dæmon can help guide the owner through a space (for example, a museum, a garden or a shopping center) by playing games or just providing extra information about the space like a guided tour.

The spaces where the dæmon can be seen have to have a video camera network and a wireless network installed since interaction depends on the recognition of the owner's actions from the video stream captured at the space and subsequent wireless communication of the dæmon's reaction to the owner's MagicLight.

MagicLight has three main components: the hardware device, the dæmons and the vuSpot infrastructure used in the implementation of the application. These three components combined bring the dæmon to real life and are described in the next Sections.

4.2.1 The Device

MagicLight is to be used with a mobile device that has projection functionality. In our case this device was obtained by connecting physically a mobile device to a pico-projector.

To implement the system we used an iPhone 4 (with 480x320 pixels resolution) connected to a MicroVision SHOWWX+ [Mice] portable HDMI laser pico projector (see Figure 4.16). The device has a total weight of 290gr (0,64 pounds) from which half belongs to the iPhone. Since the pico projector works using a laser beam, which is focus free, the projection will

Figure 4.16: MagicLight: The device.

Figure 4.17: Example of Dæmons' reactions (left to right: sad, happy, very happy, relaxed, very sad, angry, dizzy, bored/sleepy, do not know/do not understand).

always be sharp on all surfaces.

4.2.2 The Dæmons

Dæmons mimic the appearance and behaviour of the animal they represent. Several degrees of interaction with their owners can be seen in literature and movies. In "The Golden Compass" movie, they are able to communicate with their owners and with each other and even experience the feelings of their owner and vice-versa. In "Harry Potter" they are companions that help the owner by performing tasks.

Dæmons are not regular pets. They are companions for life and do not take orders. They are the facet of the owner. The user, when creating a profile (the user profile is created in an application different then the one used to explore the space), chooses the dæmon with which she identifies the best. If the user does not possess a user profile then the default dæmon will be used. In Figure 4.17 an example of a monkey dæmon reactions can be observed.

Users are asked to answer questions about their personality (emotions and interests). After the user profile is created, the dæmon is born knowing and possessing part of the owner's personality. The following questions are examples of questions that can be asked to user to obtain personality and interest information:

- You were invited to go on an expedition to the North Pole. What do you think of that? Answer choices: 1: awesome, 2: fine, 3: too far, 4: a journey of a lifetime, 5: too dangerous.
- You happen to see someone cutting in front of you in line, such as at the movies or at the register in a store. What do you do? Answer choices: 1: confront the person, 2: be angry but do nothing, 3: doesn't bother me, 4: be annoyed but do nothing, 5: ignore the situation.
- If you would enter a room full of people you don't really know what would you do? Answer choices: 1: Introduce yourself, 2: find a familiar face and stick to them, 3: walk around until someones talks to you, 4: will never happen, 5: leave the room.
- You have been asked to speak to an audience. What is your reaction? Answer choices: 1: excited, 2: relaxed, 3: nervous, 4: panic, 5: will not do it.
- What do you enjoy the most in an art gallery? Answer choices: 1: sculptures, 2: paintings, 3: architecture, 4: other forms of art.
- Which color do you prefer? Answer choices: 1: blue, 2: orange, 3: red, 4: green, 5: yellow.

With the information submitted the reactions of the dæmon can be adapted to the owner's personality. For example, a more calm person will possess a more calm and relaxed dæmon and a more stressed person will possess a dæmon that will be angry more often and more easily. The dæmon will also guide the owner through the space according to the interests submitted. This is achieved by manipulating the behaviour using probabilities inferred from the answers of the user.
The dæmon will also guide the owner through the space according to the interests of the owner. A user that enjoys examining the architecture of a space will be guided to details that a user that prefers paintings will, probably, not be interested in.

4.2.3 Application Description

To test MagicLight, we created a version of the "hot and cold" game. Three images are showed to the user and the user is asked which one is the most appealing. Three empty picture frames were hanged on the wall. Each frame belongs to only one of the images. The user does not know which frame belongs to which image (see Figure 4.18).
A MagicLight device is given to the user and the user has to guess which picture frame belongs to the image previously chosen.
The dæmon knows which one is the chosen one. For the user not to make a random guess she needs to interact with the dæmon to get more information. The user can determine if she is moving in the right direction from the dæmon reaction's.

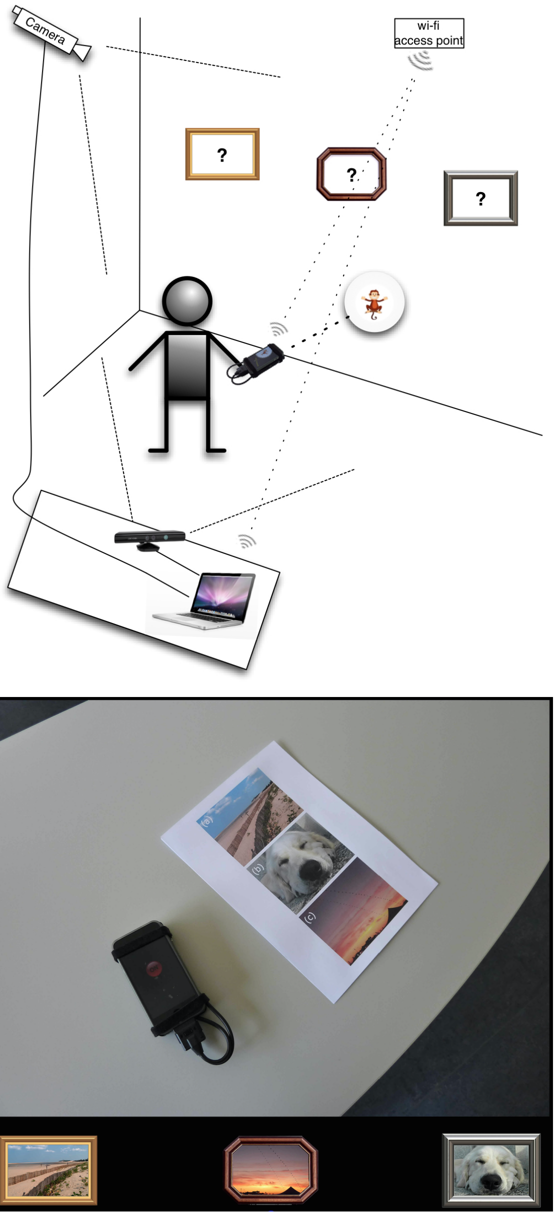


Figure 4.18: Setup.

4.2.4 Implementation

The MagicLight application was built on top of the vuSpot infrastructure. Its architecture can be observed in Appendix A.
It was created without the use of the virtual tool proposed in Section 3.3, which means that it was created by a person with programming experience. The results of the tests were used to create the virtual tool and the other applications proposed in Chapter 3.
Nevertheless, a simplest version of this game was later designed and created using the proposed applications to verify their behaviour. The description files generated by the XploreBuilder application can be observed in Appendix C and the design in Appendix D. The interactions with the dæmon were successfully designed and created.
Interactions can be occur using the MagicLight by actuating the accelerometer or compass or with gestures that are captured on video by the video camera network:

- The dæmon will relax if the user does not move and does not actuate any sensor. After some time (that depends on the dæmon's personality) being still, a relaxed dæmon is projected.
- To make the dæmon go to sleep he has to be relaxed. As for the relaxed dæmon, this can be accomplished by not moving and not actuating any sensor. After some time (that depends on the dæmon's personality), being relaxed and being still, a sleeping dæmon is projected.
- If the user shakes the MagicLight the dæmon gets dizzy.
- If the user shakes the MagicLight for a long time (depending on the the dæmon's personality) the dæmon gets angry.
- If the user puts her hand in the air or finds the correct image, the dæmon laughs.
- If the user moves away from the picture frames stepping back (i.e., moving in the direction of the Microsoft Kinect camera), the dæmon will be confused (the user is deviating from the objective).
- If the user waves goodbye to the dæmon, the dæmon will be sad.
- If the user steps away from the cameras view range, the dæmon will cry.

The user can determine if she is moving in the right direction from the dæmon reaction's (this is a hot and cold game with a twist). The location of the user also induces reactions:

- If the dæmon is projected left, centre or right in the display area then he is trying to tell the user to move in that corresponding direction.
- If the user moves away from the picture frames stepping back, the dæmon is confused.

- If the user steps away from the cameras view range, a crying dæmon is projected.
- If the user is moving in the correct direction the dæmon is happy.

If the dæmon is confused or crying, even if the user actuates the mobile device sensors in order to get another reaction, his reaction will not change until the user steps forward in the direction of the picture frames or back into camera range, respectively.
According to the user's action recognised, the user's input and the user's location, a message is sent to the mobile device with the indication of how the dæmon should react. The mobile device client receives the message and the information manager parses it obtaining the information needed to generate the dæmon response to be projected.

4.2.5 Evaluation

In MagicLight we want the dæmon to react to the owner's actions and to lead the owner to perform some actions. We created a scenario to explore the interaction between the user and the dæmon. We conducted a simple study to assess if projecting the dæmon is a good form of visualising it and if, using the existing modules of the infrastructure, we are able to create an application that uses a pico-projector to project a virtual character and interaction with it is successful.

4.2.5.1 Participants

MagicLight was tested by sixteen users. Their ages ranged from twenty to sixty-three years old, with an average age of 36,7 years old. As for gender, five of the users were female and eleven users were male. All users own a smart phone, seven with the Android operating system and nine with the iOS operating system.

4.2.5.2 Test Setup

We created a simulation of a space that has framed pictures hanging on the walls, like in an art gallery. We are interested in a specific area of the gallery where a game to be played using MagicLight was created to entertain the visitors.
To develop the system, a video camera network was created using webcams. One of the webcams is a Microsoft Kinect camera, model 1414 [Micb] (see Figure 4.18). It is important that at least one camera per space records depth data in order for the system to identify the location of the user accurately.
Before using the MagicLight the users need to fill out a questionnaire. In this questionnaire we ask questions about their likes, interests and emotions. This information is used to create the dæmon's personality (see Section "The Dæmons"). As mentioned, the dæmon's personality is an extension of the personality of the owner.
We also showed the user three images and asked which one is the most appealing.
Three empty picture frames were hanged on the wall. Each picture frame belongs to only one of the images. The user doesn't know which picture frame belongs to which image.

A MagicLight device (see Section "The Device") is given to the user and the user is asked to complete some tasks (see Subsection "Tasks").
A list of possible reactions was shown to the user but no information about how to obtain the reaction was given. The users were told to try to guess how to obtain the reactions by performing gestures and by using the MagicLight device.
To simplify the user test, only two gestures were associated to a dæmon reaction: "wave" and "raise hand". We wanted to infer if these gestures were natural choices for the users.
If, when finishing using MagicLight, some reactions were not obtained because the "wave" and "raise hand" gestures were not tried or were not performed correctly, a short tutorial on how to perform them was given and the user's had the chance of trying again for getting the corresponding reactions.
After using the MagicLight, the users were interviewed. Their answers were used to obtain information to evaluate the application and infrastructure.

4.2.5.3 Tasks

The users were asked to complete two tasks simultaneously.

4.2.5.4 Task 1: Guess which picture frame belongs to the image found the most appealing.

The dæmon knows which picture frame corresponds to the chosen image. For the user not to make a random guess she needs to interact with the dæmon to get more information (see Figure 4.19).
This task is mainly used to evaluate the accuracy of the user location calculated by the Location Module of the infrastructure.

4.2.5.5 Task 2: Explore the dæmon reactions.

The user is asked to interact with the dæmon and obtain the reactions described in Section 4.2.4 (see Figure 4.20) This task is used to evaluate:

- The accuracy of the results of the gesture detection algorithms of the Detection Module.
- The ability of the framework to generate a reaction from the information gathered from the mobile device sensors information and the streams of the video cameras of the video camera network present at the space.
- If performing gestures and actuating sensors is a good mean for interacting with the dæmon.

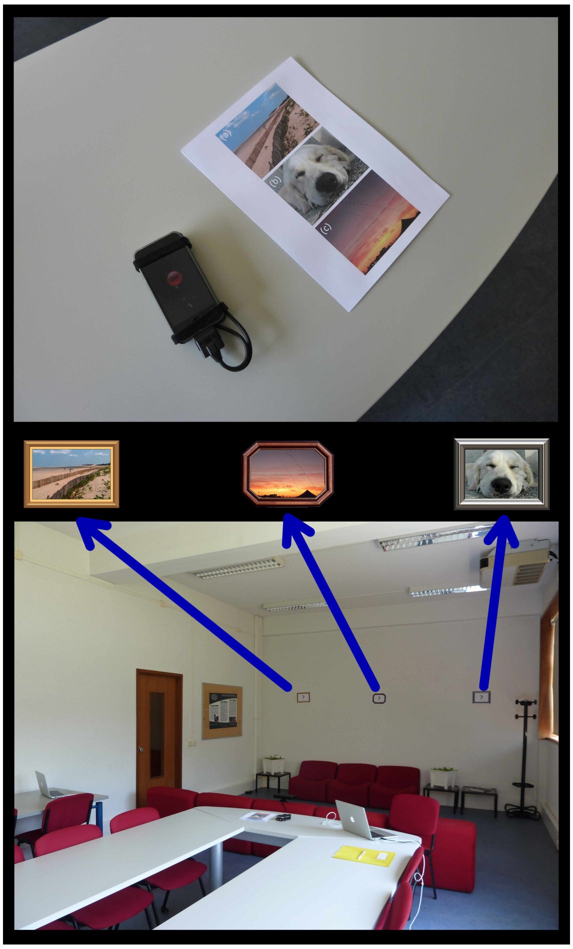


Figure 4.19: Top: images shown to the user. Middle: images corresponding to the empty frames. Bottom: empty frames representing the hidden images.

4.2.5.6 Results

All users guessed the picture frame correctly (Task 1) meaning that the dæmon was able to communicate correctly the hints about the guessed picture frame location, helping the owner complete the task successfully. Since the hints provided by the dæmon are correct we can conclude that the Location Module was calculating the location of the user correctly.

81% of the users considered easy or very easy to complete Task 1 (find the correct picture frame) and the rest had an average difficulty.

When asked to evaluate the degree of difficulty (from Very Easy to Very Difficult, see Table 4.3) in obtaining the several dæmon's reactions individually (Task 2), the laughing, confused and crying were the most difficult ones. From the remaining four, Dizzy was the reaction with the worst results because the dæmon, with some user personalities, would go from Dizzy to Angry too fast for the action to be recognised.

The user's opinion was that the dæmon's reactions made sense and had no major problems understanding the reactions and hints of the dæmon. 31% of the users (especially the older

Figure 4.20: Top: user testing MagicLight. Middle and bottom: dæmon projections.

	Obtaining a reaction				
Reactions	*Very Easy*	*Easy*	*Average*	*Difficult*	*Very Difficult*
Relaxed	56,2%	43,8%	0	0	0
Sleeping	56,2%	43,8%	0	0	0
Dizzy	56,2%	31,3%	12,5%	0	0
Angry	62,5%	37,5%	0	0	0
Laughing	56,2%	18,8%	12,5%	12,5%	0
Confused	62,5%	0	0	25,0%	12,5%
Crying	56,2%	18,8%	0	6,2%	18,8%

Table 4.3: Task 2 user tests results. Percentage of users for each reaction and degree of difficulty.

ones) had problems distinguishing between the mad and the dizzy dæmon especially when there is too much light for the projection to have the desired contrast. The dæmon image could be larger to minimise this problem or, in the future, the dæmon reaction images have to be substantially different from each other. The addition of sounds would minimise this problem.

Almost all users never used a pico projector before. All found the weight and size very acceptable and the device fun to use. We had no negative comments about the character chosen for the dæmon and the form of visualising it.

Getting reactions from the dæmon could be a consequence of either actuating the sensors of the mobile device or performing gestures. All user's said they had no problems getting the dæmon to react but when asked in particular about getting a reaction through gestures the answers diverged and 25% answered they had to repeat several times the gesture before getting a reaction. These results match the results in Table 4.3 for the reactions involving gestures (laughting, confused and crying), which were the ones considered the most difficult to obtain.

People find it difficult to interact with gestures since they are recognised using the video stream. This means that gestures have to be done in the camera range and view. Many performed the "wave" gesture with their body blocking the view meaning that their gestures would not be recognised and making it hard to get a reaction. When the test ended and after a small explanation about how to make the gestures they found it easy to get a reaction.

About a half of the user's do not have experience or have little experience using gestures to interact with systems, which means that they do not know what are the common gestures used in these systems. By observing the user's we concluded that waving was a gesture that all users tried out for getting a reaction, meaning that it is an obvious gesture to use for interacting with a virtual character such as this. The "raise hand" gesture was not immediately tried by the users. Before trying it, virtually all users, tried other gestures like tap and click.

The "raise hand" gesture was easily recognised by the system while the "wave" gesture had some problems. We concluded that the "raise hand" gesture is not an obvious gesture to use for interacting with the dæmon, if no tutorial is given first, but it is a gesture that will hardly ever not be recognised correctly while the "wave" gesture is an obvious gesture to use but has recognition problems.

The user's gave the same level of importance to reactions obtained by performing gestures as to reactions obtained by actuating the mobile device sensors, although, from the authors point of view the actions involving gestures are more difficult to be recognised and the reaction is more prone to be wrong. By having the users placing the two methods at the same level and obtaining good opinions, we concluded that using gestures is a good mean of interacting with the dæmon and the gesture detection algorithms are having correct results.

We realised that this kind of systems work better in large rooms because of the camera view and angle. The confused reaction was difficult to obtain due to the dimensions of the room. The reaction is noticed when the user steps way from the picture frames but when the distance between the cameras and the picture frames is too short, the distance limit for the reaction to take place is too close to the camera.

The questionnaire results for defining the dæmon's personality varied from user to user. For the time for the reaction to be noticed, 62,5% users considered it adequate, while 37,5% considered it too long. After filling the questionnaire and talking about the test, more than half of the users that considered it adequate confessed that they would like the reactions to be faster, meaning that more than half of the users wanted a faster reaction (not depending on the user's personality).

One conclusion we arrived at is that the users do not really care if the dæmon personality matches the owner's (or even if it is the opposite as a user suggested). They want to see many reactions and as fast as possible, independently of personality. The customisation of the dæmon should be done with other parameters and not take into account the speed of the reactions. The whole test had a duration below 5 minutes for all users, which was very satisfying.

4.2.6 Results and Discussion

After the tests we had the opportunity to talk with the users about the MagicLight experience and obtain feedback and suggestions to improve the system and infrastructure. The results were positive and we are happy with them. The users that tested the application were enthusiastic about it and allowed us to confirm that such an application has a great potential to help explore spaces.

A few users suggested to add sounds like snore, grunt, growl and laugh to make the reaction more immediate and the MagicLight more fun to use. After the user tests this functionality was added to the application. Although it was not tested with children, the users considered it would be a good application for children.

The infrastructure proved to be effective in the creation of mobile device applications involving location and gesture recognition. Although some aspects of MagicLight need to be improved (most of them not related with the infrastructure itself but with details specific for this application), applications using the infrastructure proposed can be fast and easily developed.
One of the major problems encountered was the lighting. Good conditions for the cameras are bad conditions for the projector. It is difficult to find a middle point and the outdoor use is very restricted.
More information can be projected besides the dæmon. The application could benefit from the projection of additional information about the space or other information to help the user fulfil a task.
With MagicLight we learned that with a short tutorial before using the application on how to perform the gestures, recognition of gestures is done correctly and fast.

4.3 Gone Fishing: Using Pico Projectors to Explore Other Worlds

With Gone Fishing we propose a concept of exploring virtual worlds using pico projectors. A virtual world can be described as a computer-based simulated environment [Bar03] that contains virtual characters and objects. Many definitions have been suggested over the years but what they all have in common is that the fundamental attraction of a virtual world is what it holds inside. The suspense and excitation of exploring it and taking a step towards the unknown. Pico projectors resemble flashlights, one item every explorer must have. The proposed concept consists of using the pico projector as a flashlight to explore the virtual world, showing it through its beam of light.
To demonstrate the concept we implemented a game that makes the pico projector a potential tool used to play games that provide interactive experiences while exploring virtual spaces: Gone Fishing. Our objectives are: (1) to evaluate if the vuSpot infrastructure is adequate for creating such games and (2) to explore new forms of interaction and social experiences using mobile devices and pico projectors.
An activity mostly seen performed by children but that we tend to bring with us into adulthood is collecting. The feeling of having completed a collection, possess an item others do not have or have the items needed for a certain arrangement to show off drives many people to play games like Zynga's social gaming phenomenon FarmVille [Zyna], where items need to be collected and arranged to get to the next level.
With Gone Fishing users can explore the seabed, observing virtual elements, such as fauna, flora and items that can be used as decorations of an aquarium. The pico projector projects the area of the seabed the player is pointing at. To explore the whole virtual world, the player needs to move around the projection to uncover it. Since the fauna elements have different characteristics, they move around the seabed with different speed and direction.

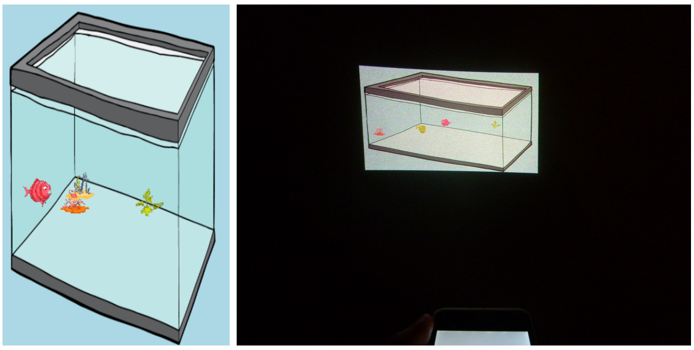


Figure 4.21: User's aquarium. Left: using the mobile device display. Right: using a pico projector.

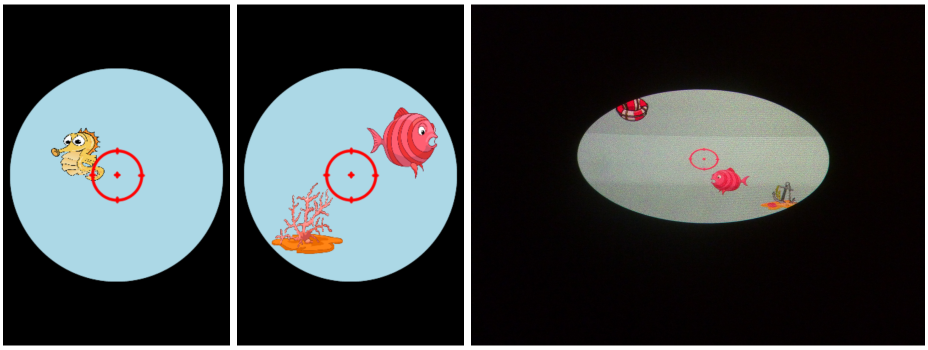


Figure 4.22: Exploring the seabed. Left and Center: using the mobile device display. Right: using a pico projector.

The degree of difficulty of finding and capturing them depends on those characteristics. The game can be played by collecting fauna, flora and decorations or, if collecting is not an activity the user likes, the user can relax while watching the fish swim around the flora and decorations.
Each user can decorate a personal aquarium. The elements to place in the aquarium have to be captured from the seabed.

4.3.1 Application Description

Gone Fishing is a game where the player explores the ocean floor and collects underwater specimens. This game also takes place at a room of a space that has a video camera and a wi-fi network installed. The pico projector attached to the mobile device projects a virtual sea bed over the room.
Each user owns an aquarium (see Figure 4.21). Fish, flora and decorations can be collected and exchanged (see Figure 4.22). To capture fish, users must pursuit and capture them by aiming at the fish and shaking the device. Since users at the same room explore the same sea bed, the wandering fish, flora and decorations are the same, which can lead to a competition between users to get the most pretty specimens (see Figure 4.23). Some animals are faster than others so capturing them is sometimes a difficult task.

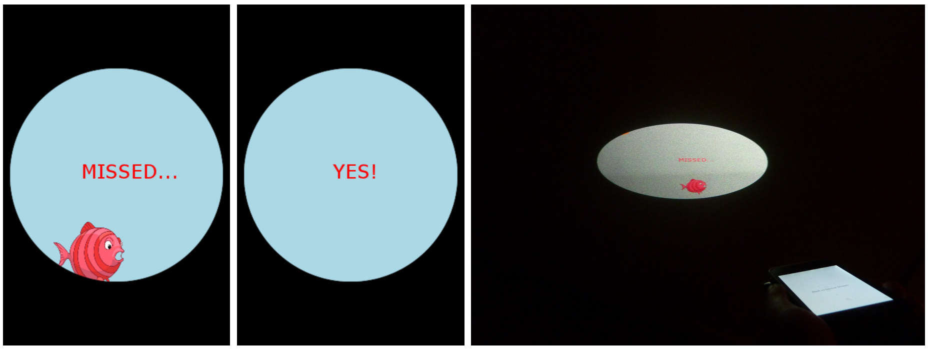


Figure 4.23: Capturing a fish. Left: missed using the mobile device display. Center: capture successful using the mobile device display. Right: missed using a pico projector.

Objects and living beings can be released to give room for new specimens or to allow other users to capture them. The personal aquarium is always available for visualisation by pointing the projection at the floor. To exchange specimens users need to release them in order for the other user to capture them.

4.3.2 Implementation

The game was designed and implemented on top of vuSpot. It was implemented in C++ using openFrameworks. In the game proposed, we used pico projectors attached to mobile devices. To implement the system we used an iPhone 4 smartphone (with 480x320 pixels resolution) connected to a MicroVision SHOWWX+ [60] portable HDMI laser pico projector (see Figure 5.10). The device has a total weight of 290gr (0,64 pounds) from which half belongs to the iPhone. Since the pico projector works using a laser beam, which is focus free, the projection will always be sharp on all surfaces.

The user performs actions using the mobile device. These actions actuate the mobile device sensors (accelerometer and gyroscope). The data from the sensors is sent to the server. The information to be displayed is generated according to the information obtained by the Location Module and mobile device sensors. This information is sent to the mobile device and displayed by the pico projector. All information is stored at the system's database.

The main difficulties of the system are determining the user's location in the room and determining to where the user is pointing the projection in order for the correct area of visible seabed to be displayed.

The video streams from the existing video camera networks are used to obtain information about the user's location (Location Module). To ensure that it is accurate, one of the cameras of the video camera network is a Microsoft Kinect Camera [Micb] model 1414. With this camera, depth information is available making it easier to locate the user.

The landscape (the virtual seabed) show static (like aquarium decorations or sea plants) and moving elements (like fish). When the landscape is generated, information about the elements is stored. Moving elements contain more information than the static. They also need information about speed and direction of movement. Moving elements do not exit

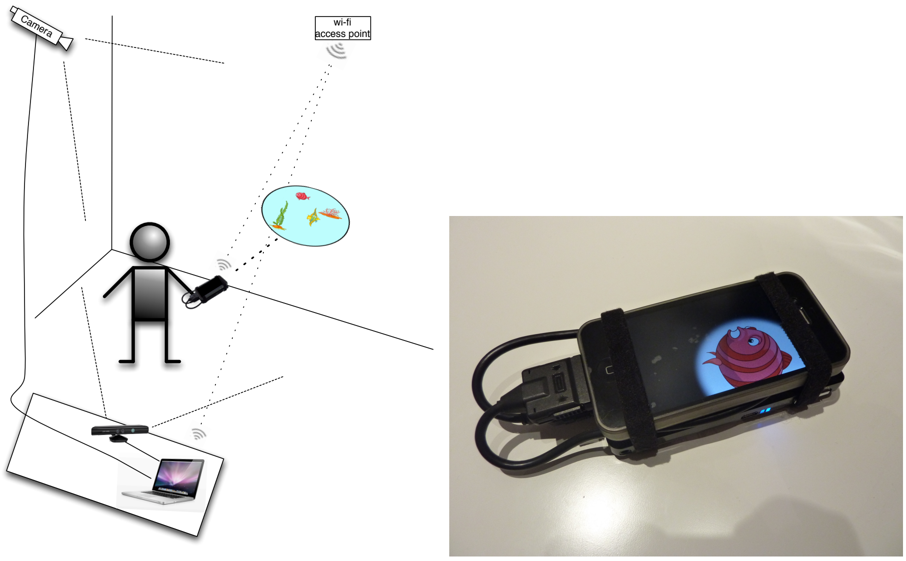


Figure 4.24: Left: the set. Right: the device.

the landscape. If the landscape is just a wall, when they reach the border they turn around. If the landscape is a whole room they wander around the room.
To keep the landscape up-to-date, the information of the elements is updated in a configurable frequency or by demand. This allows for the location of the moving elements to change in time and to change their direction randomly. By doing this, the movement of the elements is more interesting since it becomes unpredictable.
Gone Fishing was created without the use of the virtual tool proposed in Section 3.3, which means that it was created by a person with programming experience. The results of the tests were used to create the virtual tool and the other applications proposed in Chapter 3. Nevertheless, a full version of this game was later designed and created using the proposed applications to verify their behaviour. The description files generated by the XploreBuilder application can be observed in Appendix C and the design in Appendix D. The resulting game had the same functionalities than the one programmed.

4.3.3 Evaluation

We conducted a simple study to assess if exploring a virtual space using projections from a mobile device is a good form of visualising it and if, using the existing modules of the vuSpot infrastructure, we are able to create an application that uses a pico projector to project a virtual world and interaction with it is successful.

4.3.3.1 Setup

We asked the participants to play Gone Fishing using the display of the mobile phone and afterwards using a pico projector (see Figure 4.24). This will allow us to be able to compare the two experiences.

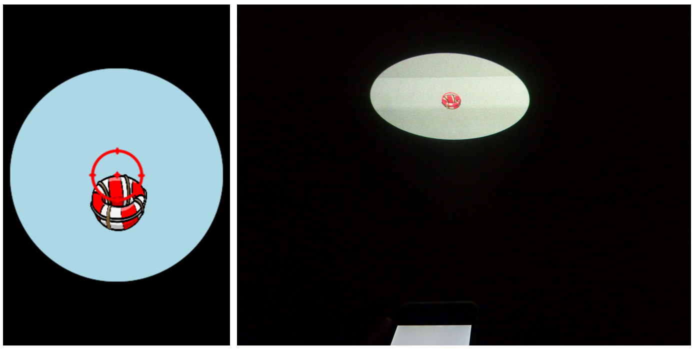


Figure 4.25: The buoy marks the center of the seabed. Left: using the mobile device display. Right: using a pico projector.

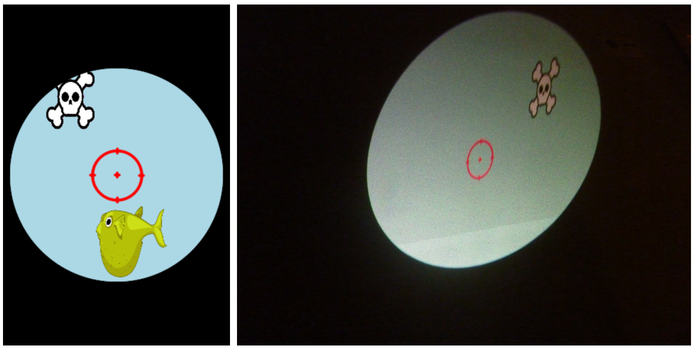


Figure 4.26: The pirate flag marks the corner of the seabed. Left: using the mobile device display. Right: using a pico projector.

For the first user tests we used a whole room (consisting of four walls) to create the seabed. When conducting the user test we soon realised that participants visualising the game using the display of the mobile device were rapidly getting lost in the seabed. Without any reference it was hard to figure out where they were. To minimise that problem we created a buoy in the location where the user "enters" (starts exploring) the seabed (see Figure 4.25). Even with the buoy the problem remained since as soon as the user turned 90 degrees in one direction she would loose its reference.
To be able to compare the experience of playing the game using the display of the mobile device and using a pico projector, we restricted the seabed to one wall. This way we gave a chance for the user to successfully explore the seabed looking at the display of the mobile device. To indicate the border of the seabed, pirate flags were placed at the corners (see Figure 4.26). Users explored the seabed being stationary at the center of the wall. After playing the game, the users were interviewed. Their answers were used to obtain information to evaluate the application, interaction and infrastructure.

4.3.3.2 Participants

The game was tested by 14 users. Their ages ranged from thirteen to sixty-four years old, with an average age of 39,2 years old. As for gender, 3 of the users were female and 11 users were male.

4.3.3.3 Results

We had a good response from the participants. All had fun and considered the game interesting and very easy to play. All participants used a pico projector less than three times before this test. They found the weight and size of the device very acceptable and easy to use.
To be able to determine if the projection of the virtual world was a good form of visualisation and if it provided means for interaction, we tested the game being visualised using the display of the mobile device and using a pico projector.
All participants agreed that playing the game using the projector was much easier. Even with the simplifications, users would get lost in the seabed when using the display. To improve the user experience, we added a shortcut (double tap on the display) to take the users back to the buoy. Users kept using it.
We realised that there is an error from the gyroscope sensor of the mobile device that is larger when movements are sudden and erratic. This influences more the visualisation on the display of the mobile device then the visualisation using a projector because the gestures performed are less fluid and are faster. With the projection the accumulated positioning error over time was almost negligible but when visualising using the display of the mobile device it made the buoy be decentered. When it happened, it was corrected using the double tap shortcut.
With the projection we realised that the wall served as a spatial reference and the users did not get lost. There was no need for shortcuts and they could navigate easily throughout the seabed.
Only one user had difficulty in visualising the aquarium. The gesture was not being performed correctly.
Catching items also posed no difficulty in both scenarios but 85,7% of the users considered more difficult to keep the aim at the target while using the display than when using the projector. The other 14,3% considered it the same in both scenarios.

4.3.4 Haunted House

In this Section, we describe another game, called Haunted House, that makes the pico projector a potential tool used to play games that explore spaces. We did not perform user tests with Haunted House. The objectives of this game are similar to the ones in Gone Fishing:

1. To evaluate if the vuSpot infrastructure is adequate for creating such games.

2. To explore new forms of interaction and social experiences using mobile devices and pico projectors.

3. Provide one more example of an interactive experience application that can be fully designed and created using the proposed tools in Chapter 3.

People associate the concept of a haunted house to dark dusty old houses. A haunted house is expected to have a ghost living in it but we can also find other beings in literature such as spiders, mice and bats. Thinking about haunted houses provokes a feeling of unease on the most impressionable people and a feeling of nervous expectation on the adventurers. While some might run away from them, others are glad to run into and explore the mysteries of these houses.

When using a mobile device with projecting functionality, we can project the interior of a virtual haunted house over a room. Pico projectors require a dark room in order for the projection to have an acceptable quality, which is one of the classic characteristics of a haunted house.

Exploring haunted houses at fairs is normally done by groups of friends. Confronting the fear along with our friends makes it become fun. The suspense and excitation make way for social interaction to take place when people hold on to each other before taking a step towards the unknown.

When using pico projectors only a part of the scenery is showed at a time and the player is forced to move around to unveil the rest of the scenery that can contain scary elements or hidden secrets.

Haunted house is a game, best to be played in groups, in a room using a pico projector, that aims at making the users experience the same feelings as visitors of a haunted house at a fair. Interactions between elements of the group might be induced by the game. We aim at exploring new forms of social interaction using pico projectors and groups of people playing the game.

The game takes place at a room of a space with a video camera and a wi-fi network installed. The lights of the room are dimmed so the projection has a better quality but it is not completely dark since the users have to be able to see where they are walking. The haunted house is projected on the walls of the room by a pico projector attached to a mobile device (see Figure 4.27). As in MagicLight (see Section 4.2) and Gone Fishing (see Section 4.3), to implement the system we used an iPhone 4 (with 480x320 pixels resolution) connected to a MicroVision SHOWWX+ [60] portable HDMI laser pico projector (see Figure 4.27). The device has a total weight of 290gr (0,64 pounds) from which half belongs to the iPhone. Since the pico-projector works using a laser beam, no focus is required and therefore the projection will always be sharp on all surfaces.

The player only sees a small portion of the interior of the haunted house room and has to move around the space pointing at the walls and objects to reveal the whole room. The image of an aim is projected so the user has a reference for the centre of the projection to facilitate interaction. Objects present at the haunted room may have secrets. To interact with the virtual objects the player must perform gestures with the mobile device, like shake or draw a circle in the air. Frightening sounds can be heard while interacting with

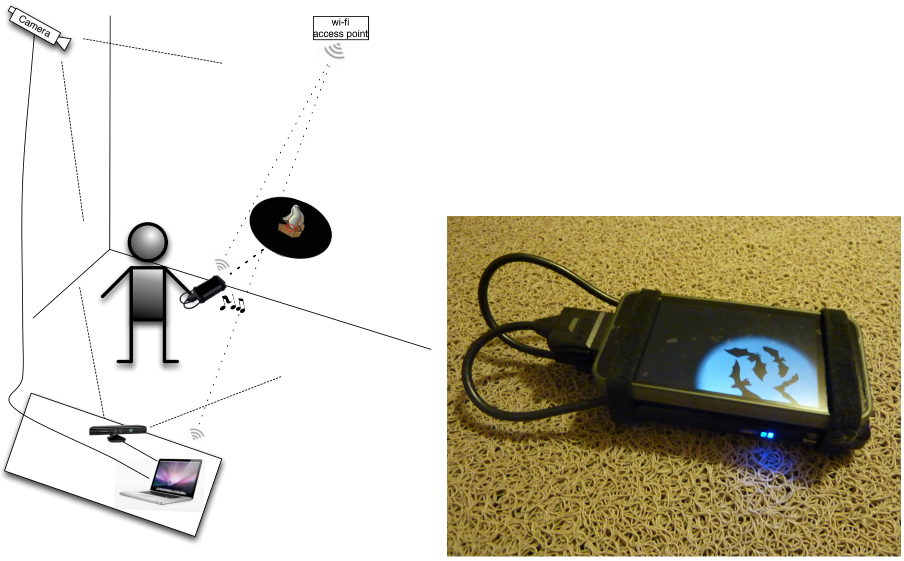


Figure 4.27: Left: the set. Right: the device.

Figure 4.28: Virtual objects and characters. Left: interaction with a virtual object (chest). Right top: bats. Right bottom: mice.

the virtual objects and characters.
When shaking the device while pointing at a treasure chest, a scream will be heard, the chest will open and a ghost will come out (see Figure 4.28). Virtual characters like bats or mice might appear suddenly to scare the player (see Figure 4.28). The objective of the game is to provide entertainment by creating suspense and, ideally, by scaring the users. Haunted House was created without the use of the virtual tool proposed in Section 3.3, which means that it was created by a person with programming experience. Nevertheless,

a full version of this game was later designed and created using the proposed applications to verify their behaviour. The description files generated by the XploreBuilder application can be observed in Appendix C and the design in Appendix D. The conclusions drawn while designing and creating this game were used to improve the visual tool and the other applications proposed in Chapter 3.

4.3.5 Results and Discussion

We propose a concept of using pico projectors to explore virtual worlds. We created two games, called Gone Fishing and Haunted House, to demonstrate the concept. They are built on top of vuSpot, an infrastructure that facilitates the creation of augmented reality games using mobile devices. The infrastructure is composed by modules that can be combined to create different kinds of applications. It proved to be adequate for this kind of applications.

Gone Fishing can be played in several ways: just to explore the sea bed, as a collecting items game or as a competition between players. Social interaction can take place while capturing specimens and comparing aquariums. People are more likely to explore a Haunted House in a group. Haunted House is to be played solo or in groups and aims at provoking emotions of fear, suspense and curiosity while exploring a haunted house room. When played in group, this game may induce social interaction between the members of the group.

Although the games do not have a score system, it can be easily implemented by attributing scores to the elements the user captures in Gone Fishing and by attributing scores to the objects the user interacts with in Haunted House.

We tested Gone Fishing and the results were promising. We tested the game being visualised using the display of the mobile device and using a pico projector. Even with some simplifications to facilitate the visualisation of the game using the display of the mobile device, the users were unanimous in saying that it was much easier to play the game while projected. Their opinion was also that it was much more interesting and fun.

We can conclude that using pico projectors in applications that explore virtual spaces is an advantage and that it provides a better user experience.

We also concluded that with the infrastructure vuSpot we are able to create games that use pico projectors and mobile devices to explore virtual spaces. The infrastructure behaved correctly and posed no problems.

When using pico projectors, restrictions on information display are minimised since a large area for displaying information becomes available. Pico projectors combined with mobile devices equipped with sensors have a potential application in gaming and are a innovative way of exploring virtual spaces.

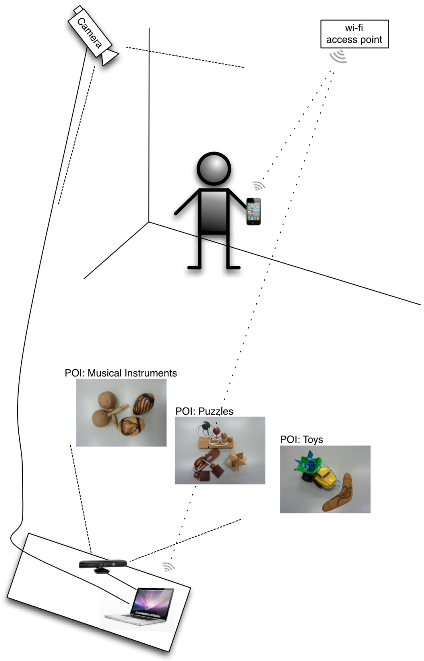


Figure 4.29: Setup.

4.4 Toy Exhibition

The Toy Exhibition was created to study how people design and use simple interactive experience applications. It is an example of the use of the four components of this research working together: vuSpot, urSpace, XploreBuilder and XploreGrammar. An exhibition space was created with a show of toys. Users were asked to design and play mini-games at the space. By observing the users designing (and creating) and playing a designed game, we are able to draw conclusions about the behaviour and performance of the tools.

4.4.1 Application Description

For the Toy Exhibition, a space was created. To simulate it, a video camera network was created using webcams. One of the webcams is a Microsoft Kinect camera, model 1414 [Micb] (see Figure 4.29). As mentioned, it is important that at least one camera per space records depth data in order for the system to identify the location of the user accurately.
We configured beforehand the Toy Exhibition space in the XploreBuilder tool (see Section 3.3, this step is only done once and is used to create all applications). Since, the interactive experiences to design were mini-games, the configuration was kept simple but with enough building blocks to create very different applications (see space description file in Appendix C).
Three Points Of Interest were created: a set of toys, a set of puzzles and a set of musical instruments. The gestures available at the space to create the Action Blocks were: wave, click and raise hand. The reactions available consisted of displaying text (three predefined sentences: "Hello!!, "How are you?" and "See you soon!"), images (two predefined images,

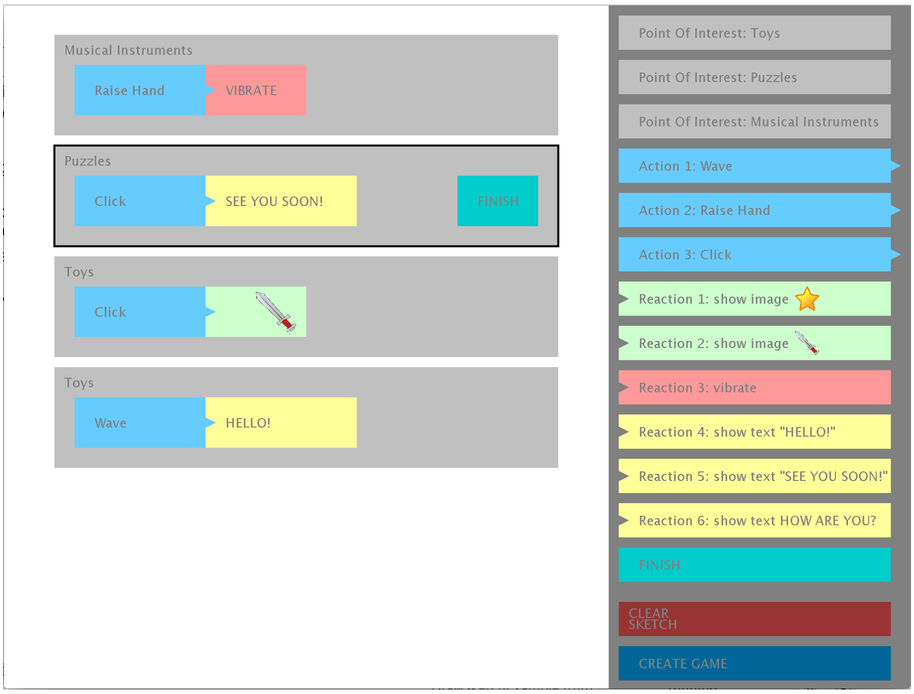


Figure 4.30: XploreBuilder simplified interface.

a sword and a star) or making the mobile device vibrate.
To simplify the creation of interactive experiences at the Toy Exhibition, and since there were only a few building blocks available that could fit in the application's window without the use of menus, the interface was simplified so all information about building blocks was available. The building blocks available are predefined according to the space (see Figure 4.30) initial configuration. We described the interactive experience applications using a grammar based representation scheme (see Section 3.2). Although the grammar allows for more complex designs (using composite actions and reactions and time blocks) we focused our attention on the more simple sequences in order to be able to write a game (mini-game) for a small space and with a short duration so it could be used in our user tests.
The representation of the grammar for the Toy Exhibition can be observed in Appendix C. We restricted the space exploration to the creation of a mini-game that is composed of several sequences of action/reaction at three Points Of Interest and a finish sequence.

```
{
MUSICAL_INSTRUMENTS: RAISE_HAND -> VIBRATE,
TOYS: CLICK -> SHOW_IMAGE: sword.jpg,
TOYS: WAVE -> SHOW_TEXT: Hello!
},
PUZZLES: CLICK -> SHOW_TEXT: See you soon!
```

Listing 4.1: Representation of a game.
For the game in Figure 4.30 we would obtain the description file in Listing 4.1. Each grey POI Block with its Action and Reaction Blocks is converted into a sequence. The Puzzles

POI Block, which has the Finish Block, will be the Finish sequence.

4.4.2 Evaluation

Our objectives for this test are:

- Objective 1: Identify the difficulties of designing an interactive experience for a given space and determine which features are important to have available while designing that interactive experience.
- Objective 2: Evaluate if the created application is what the designer had in mind while designing it, that is, if the system was able to reproduce the interactive experience application the user designed.
- Objective 3: Evaluate if the infrastructure, which urSpace is built upon, is adequate for the design, creation and use of the interactive experiences and if the gestures are correctly recognised and the reactions to those gestures transmitted correctly to the user.

4.4.2.1 Participants

The test involved 17 persons. Their ages ranged from 18 to 62 years old, with an average age of 30,4 years old. As for gender, 4 of the participants were female and 13 participants were male. Half of the users have little or no programming experience.
The target audience for designing the interactive experience using the XploreBuilder tool is space owners/managers, curators or the general public if it is made available. The target audience for using the interactive experience application is anybody who can perform gestures and use a mobile device. Our aim is that both designing (and creating) and using the interactive experience application is done by the general public, meaning that a person with little or no experience in providing means to visit spaces and application design is able to create successfully an interactive experience application.
It is also important to make user tests with users with programming experience, since these users know what to expect from this type of applications and can give a different kind of feedback. They gave valuable suggestions on what interactions could be created (although most of the participants are not regular video game players, many of the suggestions were related to games) and what actions would be interesting to be recognised to be able to play more complex games.

4.4.2.2 Tasks

An explanation about the Toy Exhibition space and its contents was given so users could imagine an interactive experience to be used at that space. Afterwards the participants

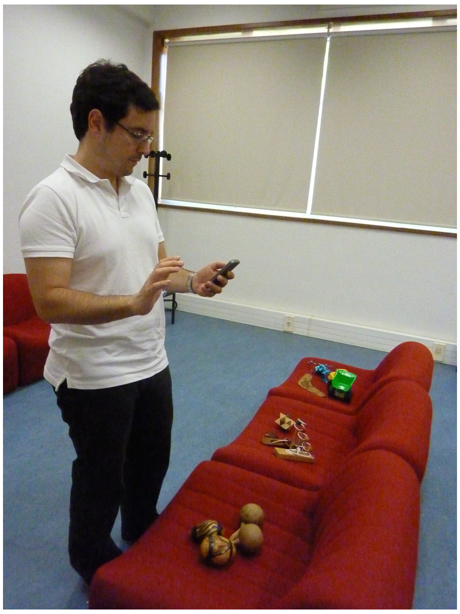


Figure 4.31: User playing the game he created.

were asked to complete two consecutive tasks:

Task 1 - Design a Game: The participants were asked to design a mini-game using the XploreBuilder visual tool for the Toy Exhibition using any combination of POI, action and reactions available at this space. The XploreBuilder interface was briefly explained. An introduction to how to describe an interaction and what are Points Of Interest, Actions and Reactions was given.

Task 2 - Play the Designed Game: The participants were asked to play the designed mini-game at the Toy Exhibition space (see Figure 4.31).

After completing the tasks, the participants were interviewed. Their answers were used to obtain information to evaluate the applications behaviour and the infrastructure performance.

4.4.2.3 Results

We organised the results according to our three objectives.

Objective 1: Identify the difficulties of designing an interactive experience for a given space and determine which features are important to have available while designing that interactive experience.

We observed that the participants were very fast at understanding the concept of an interaction sequence and what the several building blocks represented. Few doubts had to be answered. This was in part due to the fact that the XploreBuilder interface was designed taking in consideration the knowledge gained from the strategies and techniques of other successful video tools.

Designing a game took less then 4 minutes for all participants. This was in part due to the fact that the participants created very simple games because they were trying to use all the gestures at different Points of Interest and not creating more complex games.

Most of the difficulties took place when the users tried to correct already designed interactions. The application uses the mouse double-click to delete blocks and it works well, deleting a block at a time. Re-using the blocks of the interaction sequence that were left undeleted and making sure they were not left without an association to an interaction was a problem.

Almost 90 % of the participants considered designing a game with XploreBuilder easy or very easy.

From a set of 27 words (boring, pleasant, tiring, complex, comfortable, confusing, convenient, hard, fun, thrilling, easy-to-use, frustrating, impressive, innovative, intuitive, annoying, motivating, too technical, great, bad, satisfactory, worthless, simple, sophisticated, stressful, useful, addictive), the participants chose 15 to describe the experience of creating their own game using XploreBuilder. All the words chosen have a positive meaning, being the most used: easy-to-use (12 times), pleasant (10 times), intuitive (8 times), fun (7 times) and simple (7 times).

We reached the conclusion that this form of creating the space exploration is very well accepted being easy and fast to create interactive applications to help explore a space.

Objective 2: Evaluate if the system was able to reproduce the interactive experience application the user designed.

The objective of urSpace is to create fast and easily applications to help explore a space. We chose to ask the designer to also play it because the designer is the only one that knows exactly what functionalities should be offered in the game and we wanted to evaluate if the designed application matched the created application.

Although it was explained at the beginning of the user test that the test consisted in designing a game and afterwards playing it, many participants were surprised when asked to play the game right after designing it. Participant's thought that it would take some time to create the game. We learned that being able to immediately use the application after designing it is a major advantage of the system and it brings satisfaction to its designer. When starting to play the game they created, participants were highly motivated and happy to be playing that specific game.

The users were very pleased for being able to play the game they designed and the game met completely or most of the expectations of 82% of the users. The rest of the users commented that they were expecting that playing the mini-game would be more complex (although the game reflected what they designed).

One conclusion we reached was that game design benefits from having the designer play the game to gain a complete view of the interactions and their fluidness in the game. After playing the game many users asked if they could go back to the design and add more interactions. This was out of the scope of the user test but, since the XploreBuilder provides means to open an already created experience, updates and enhancements can be done at any time.

Objective 3: Evaluate if the infrastructure, which urSpace is built upon, is adequate for the design, creation and use of the interactive experiences and if the gestures are correctly recognised and the reactions to those gestures transmitted correctly to the user.

Our last objective was to evaluate if urSpace works correctly with the The Infrastructure. Almost all gestures performed by the participants were immediately recognised and the mobile device displayed immediately the corresponding reaction (or vibrated). Participants not familiar with video processing and stereo cameras (about two thirds) were amazed that they were able to interact with the objects at the exhibition by only performing gestures in the air. This is an indicator that this kind of technology is interesting to have in spaces that contain objects for the general public to visit.
Three participants had difficulties with the wave gesture. When playing the game the gestures performed tend to be artificial. The wave gesture is a short and rapid movement of the hand horizontally. The participants were performing the wave gesture by moving the hand in a long slow movement almost from one side of the body to the other. This caused the gesture not to be recognised. When we told them to wave like they would wave to a friend saying goodbye the gesture was immediately recognised.
One participant had problems with the click gesture. He was performing the gesture sideways instead of in front of him.
We realised that the infrastructure has problems recognising gestures done slightly differently then expected but it is also difficult to program all variations of gestures. Since algorithms for gesture recognition can de added or upgraded at any time in The Infrastructure, we hope that research in this area will result in better algorithms.
One suggestion from the participants was to give a demonstration of how the gestures are supposed to be performed before playing the game. Either on site before the user tests or by, for example, playing a video tutorial before playing the game.
Overall the detection and location modules worked correctly and we are happy with the results. urSpace worked correctly in conjunction with vuSpot and XploreBuilder tool.
The set of existing gestures (actions) should be expanded. The gestures available at the user tests were found appropriate for using in this type of applications but not enough to satisfy the users needs. The users craved for more actions and wanted to try all. The more actions exist, the larger is the number of possible games to design. Although we commented that, in fact, the set of existing gestures is larger than what was shown and we described all the available gestures, users wanted more. They suggested to use gestures and reactions similar to those in Role Playing Games like Diablo [Blia] and The World Of

Warcraft [Blib] or the sandbox game MineCraft [Moj].

4.4.3 Results and Discussion

The work proposed aims at simplifying the design and development of applications that use space exploration and interaction with a space as main features. After the user tests we had the opportunity to talk with the users about the system and obtain feedback and suggestions to improve the applications urSpace and XploreBuilder and vuSpot.
We evaluated how users used the XploreBuilder visual tool and evaluated its ease of use. The results showed that the participants found the application easy to learn and use. The words: easy-to-use, pleasant, intuitive, fun and simple were used to describe the experience of designing a game using XporeBuilder.
They also thought that XploreBuilder is simple to use and the games created meet the creator's expectations. Although we removed the menus to simplify the application's interface, the users commented that they would prefer to see the blocks organised in menus.
As for vuSpot, the results were positive. vuSpot works well in conjunction with the other tools and its modules had the expected behaviour.
Although we did not test the created games with children, the system is suitable for the creation of educational interactive games. These games can be mini-games where the interaction with the objects will make the children learn more about them while having fun.
One of the main advantages of this system is that it does not require specific knowledge (including programming languages) to create an application. By using a visual tool the design of the application is simplified and the development is mostly automatic and immediate.
The games generated were very simple but users were amazed that they could immediately play the game they created.
The applications for this system include collaborative gaming where actions trigger reaction on other players and players need to work together to achieve goals. When looking at Role Playing Games (RPG) many interactions between users can be found and can be brought to this system. Players playing the same game might work together or might fight with each other. RPG are the most complex games to be created with this system. Two frequent game players were very interested in the system to create customised RPGs. With enough actions and reactions available, many different RPG can be created for the same space and users (other than the curador or the owner) can take turns at creating the game and playing it.
Another application for this system involves a public ambient display instead of a mobile phone to display the reactions. Players can work as teams and information about the game progress is visualised at the ambient display. This version of a game makes it easier for the players to play due to the fact that they do not need to install an application on their

mobile device. This is specially interesting when creating games for visiting schools and special events.

Although the system is intended to be used at spaces people visit, a user suggested to use it at home to command appliances without the use of the mobile device. As long as there is a video camera network available, actions can be recognised. The reactions, in this case, are not information to display but the trigger of conditions to operate devices.

Some actions might involve facial expressions but this is only possible if there is a camera very close to the user or if there is a camera dedicated to this action were the users can go to and interact.

A few users suggested that we could use tangible interfaces [Hor10]. The user could physically interact with the objects, for example by putting together pieces of a puzzle, playing instruments or perform an action with an object. This is a very interesting feature to include in the system as long as the interaction is possible to be recognised by processing the video camera streams. Many of this type of interactions use sensors or tags and we do not want to use them. One of the objectives of vuSpot is not to require other devices installed at the space other than the cameras.

CHAPTER 5

Conclusions and Future Work

This chapter presents a summary of the developed work and its findings as well as future research directions.

5.1 Research Summary and Findings

The main goal of this research was to study solutions for creating interactive applications to help explore a space.
These solutions had to allow an end-user with little or no experience in using programming languages, to design, create and maintain the applications to address the research question 2: "How can an end-user with little or no programming experience design and create augmented reality applications to help explore a space?". The end-user can be a space manager/owner, a curator, a teacher or even the general public if the solution is made available.
To be able to answer the research question 3: "How to create an infrastructure to augment a space and create interactive experiences, taking advantage of a video camera network without adding other objects or hardware?", we assume that a video camera network and a wireless network are present at the space (or that can be installed) and that the only additions to the space (if any) are extra video cameras.
With this requirement in mind we studied how to interact with virtual characters and objects to be able to answer research question 1: "How to interact with virtual characters and objects present at a space?".
The solution presented is composed of an infrastructure, a grammar and two tools. The infrastructure, called vuSpot (described in Section 3.1), is modular and can be adapted to the space characteristics provided that the space meets the requirement of having available a video camera and a wireless network. The modules are used according to the objective of the application to create. The algorithms and techniques used in the modules had to

be researched to understand their needs for input and their form of output in order to make sure the modules can be updated and new algorithms and techniques added as technology evolves. This infrastructure uses the video stream from the cameras present at the space to provide means to create interactive experiences while exploring that space.
We chose to use gestures as a form of interaction and studied if they are adequate for interacting with virtual characters. Since we want to take advantage of the video camera network present at the space, interaction using gestures that are recognised in the video streams was a natural choice. Other, more common, forms of interaction that are also available consist of using the input capabilities of the mobile device and actuating its sensors. By knowing what the user is doing and where, responses of the system can be generated and the mobile device application can display those responses.
Augmented reality mobile applications can be created as well as applications that use pico projectors to display information. Pico projectors possess the advantage of not having the small display restraints that mobile devices have.
The UBI, The Guardian Dragon system presented in Section 4.1 shows that it is possible to create augmented reality mobile applications using vuSpot to help explore a space. In this application users interact with UBI using gestures.
The MagicLight application presented in Section 4.2 and the Haunted House and Gone Fishing applications presented in Section 4.3, show that it is possible to create applications that use pico projectors to augment a space using the infrastructure vuSpot. Users found pico projectors an innovative, fun and easy to use way of displaying information.
MagicLight projects a virtual character that helps explore a space and, as in UBI, The Guardian Dragon, the users interact with it using gestures.
With Gone Fishing and Haunted House, we concluded that using pico projectors in applications that explore virtual spaces is an advantage and that it provides a better user experience.
By successfully creating these applications on top of the infrastructure vuSpot, we answered the research question: "How to create an infrastructure to augment a space, taking advantage of a video camera network without adding other objects or hardware?"
All these applications were programmed, using the C++ programming language and OpenCV, on top of the infrastructure vuSpot but another goal of the research is to provide means for an end-user that has little or no programming experience to create interactive applications.
XploreBuilder, described in Section 3.3, is a visual tool used to design interactive experiences by assembling building blocks. It generates the interactive experience description (that complies with XploreDescription, described in Section 3.2, which is a language grammar to describe interactions based on gestures and user's actions) and the space description. Little or no experience in programming is needed to use XploreDescription. urSpace, described in Section 3.4, is built on top of vuSpot. The generated descriptions are parsed by urSpace, that creates the interactive experience application using vuSpot modules. This interactive application can be immediately

accessed using the user's mobile device.

The combination of the XploreDescription, XploreBuilder, urSpace and vuSpot provide means to design and create interactive experiences, such as augmented reality applications, with little or no programming languages knowledge, answering the research question 2: "How can an end-user with little or no programming experience design and create augmented reality applications to help explore a space?".
UBI, The Guardian Dragon, MagicLight and the Toy Exhibition application use gestures to interact with virtual characters. We evaluated how users perceive the interaction. The results showed that the participants are fond of interacting with them. We evaluated if performing gestures was an easy and adequate form of interaction to answer the research question 1: "How to interact with virtual characters and objects present at a space?". The results were positive and the users were able to interact and obtain reactions from the virtual characters.
The infrastructure vuSpot, on top of which the five applications were successfully built, proved to provide means for creating applications where interaction with virtual characters and objects are present, answering research question 3: "How to create an infrastructure to augment a space and create interactive experiences, taking advantage of a video camera network without adding other objects or hardware?".
By being able to replicate the programmed applications described in Chapter 4 using the XploreBuilder application together with urSpace, we showed that interactive experiences can be created at a space with little requirements (a video camera network and a wireless network need to be present or can be installed) by a person with little or no programming experience. The creation of such applications becomes fast, flexible and requires little investment or modifications to the space.
The proposed infrastructure, grammar and tools have a considerable amount of potential applications, including information systems for the general public, culture, entertainment, education, surveillance, indoor and outdoor activities and advertising.

5.2 Limitations

In this section limitations and possible improvements of the solution purposed are described.
One of the key aspects of vuSpot are the algorithms to detect and recognise people and their actions. vuSpot is as good as those algorithms are, meaning that there is always room to improve them. The better the algorithms, the better the performance of the system. vuSpot is composed by modules. This facilitates the inclusion of new algorithms or the improvement of the existing ones without disturbing the other parts of the system. As research and technology evolve improvements can be made and new algorithms can be added.
The set of recognised gestures can be enlarged to contain more complex gestures, although

the number of available gestures will always be limited and probably lower than the user's expectations. The ability to accurately recognise large sets of gestures is still an open research problem [TI+15]. Sets containing about thirty gestures or more are problematic since the recognition accuracy will tend to decrease rapidly.

As mobile device technology evolves so do the applications that use that technology. Hardware is always falling behind as software becomes more and more demanding making people crave for faster processors and larger storage. In our research we augment video streams of the space the user is at to create mobile augmented reality applications. One of the major challenges in this research was video streaming to the mobile device. Smartphones have greatly evolved in the past decade but are still far from having the desired computational power and storage capacity to use this kind of applications. Streaming video not using other means than a regular laptop and controlled by a programming language proved to be a hard task. In order to successfully deliver the augmented video to the mobile device, strategies like frame dropping were used. The results were acceptable but the delay could be eliminated using better hardware and a dedicated streaming server.

A limitation of XploreBuilder is not allowing the creation of new content visualisation layouts (see Section 3.1.3) for the reactions to visualise on the mobile device. The arrangement of the contents on the mobile device screen is defined for each information type (virtual elements, text, images, video, questions and input) which means that the each type of information is visualised in the same way in every application. A visual tool to define layouts could be created and when content is added it would have to be associated to a layout. Showing collected items would also benefit from being able to define the layout of the contents. For now the items are shown at random locations on the display area.

It would also be interesting to provide means to create new types of reactions (and their visualisation layout). The system provides the most common ones but depending on the hardware of the mobile device and on the application, some user's might feel the need to have other reactions.

While creating the applications of Section 4, that were successfully built on top of vuSpot, using the tools in Section 3 we noticed that XploreBuilder was not providing the necessary means to design them. We improved the XploreBuilder tool, the XploreDescription and urSpace to be able to create them. Nevertheless there were still two functionalities that were not implemented and that were needed in MagicLight. One is the user profile. The experience of exploring a space benefits of the user's interests being taken into consideration while creating the route or interactions.

Each application requires specific information, like, for example in MagicLight the questions asked to define the user's interests and personality features. A user profile to be given to urSpace, or XploreBuilder, needs to be generic enough to be used to create automatically the interactive application and needs to fit all applications needs. This is difficult to achieve since there is no way of knowing what information new applications will need. One possible approach is to research existing applications, make a survey of the input they need and create a standard profile where common information can be entered.

The other functionality missing is being able to define the alignment of an image in a projection. This behaviour was specific of MagicLight and we did not want to add that functionality merely for the use in MagicLight. A solution might be the use of customised layouts as described earlier or the creation of a configuration page for each block where location and dimensions of the content associated can be inserted.
As features are added to vuSpot, interactions might become more complex or new forms of interaction might emerge. In this case, the urSpace, the XploreBuilder and the XploreDescription need to be updated to reflect those new features. Also, if new blocks, such as cycles, are added to the XploreDescription, the other tools need to reflect the changes. Allowing cycles of interactions was avoided because it requires a level of algorithmic reasoning that might be difficult to find in people with no knowledge of programming languages, which are the target audience for the system.

5.3 Future Directions

This work can follow several directions. As future work, we intend to extend this research to integrate:

- Telepresence to explore spaces. Telepresence was briefly approached with UBI, The Guardian Dragon. Camera views could be chosen from a map and it was possible to visualise, on a web page, the video stream the visitor was visualising on the mobile device. We decided not to follow this path since it was drifting from the objectives. As future work it would be interesting to provide means to visit the spaces remotely. A user could follow a visitor at the space or visit the space through the point of view of the cameras. The mobile device camera of a visitor could be used to capture the visit that could be used to show the space through his points of view to a remote user in real time.

- Remote interaction. Another side of telepresence is remote interaction. Interactions like the ones that occur at the space between the visitors and the points of interest, being these real or virtual, can be performed remotely. This form of interaction has to be further developed. Gestures could be used, just like at the space, provided that the user possesses a video camera connected to the computer. The difficulty of determining the position of the user at the space has to be addressed. A remote user could be a virtual visitor with an avatar and could be seen and have a location just like a virtual character.

- Interaction between visitors or virtual characters. This was not an objective of our research but it is a natural step to take. Interaction between users of the applications would make them more interesting. In UBI, The Guardian Dragon, the virtual characters of the visitors could interact with each other. They could cooperate to fulfil a task. The same could happen in MagicLight. Just as the dæmons that inspired

MagicLight, communication and cooperation could exist. In Gone Fishing items collected could be exchanged by the users.

- Social interaction in groups. Haunted House introduced the mean for interaction in groups. Visiting a haunted house is usually done in groups and the fear and emotions associated make way for people to interact with each other. It would be interesting to study how this interaction takes place and if the application can influence it.

- Tangible and wearable interfaces. Some spaces have multitouch tables installed to offer additional information in an interactive way [Cor+10] and wearable interfaces can also be found that provide entertainment [Mar+08]. In our work we do not intend to install additional objects at the space but if other objects exist beyond video camera network hardware that we can connect to our system, these objects can improve the visiting experience. They can provide information about the user's actions or be part of the interactions. Quizzes and challenges could include the manipulation of real objects.

Bibliography

[Abo+97] G. D. Abowd, C. G. Atkeson, J. Hong, S. Long, R. Kooper, and M. Pinkerton. "Cyberguide: A Mobile Context-aware Tour Guide". In: *Wirel. Netw.* 3.5 (Oct. 1997), pp. 421–433. ISSN: 1022-0038. DOI: 10.1023/A:1019194325861. URL: http://dx.doi.org/10.1023/A:1019194325861.

[AR11] J. Aggarwal and M. Ryoo. "Human Activity Analysis: A Review". In: *ACM Comput. Surv.* 43.3 (Apr. 2011), 16:1–16:43. ISSN: 0360-0300. DOI: 10.1145/1922649.1922653. URL: http://doi.acm.org/10.1145/1922649.1922653.

[Akh+14] Z. Akhtar, N. Martinel, C. Micheloni, and G. L. Foresti. "Mobile Re-Identification Based on Local Features Analysis". In: *Proceedings of the International Conference on Distributed Smart Cameras*. ICDSC '14. Venezia Mestre, Italy: ACM, 2014, 11:1–11:6. ISBN: 978-1-4503-2925-5. DOI: 10.1145/2659021.2659037. URL: http://doi.acm.org/10.1145/2659021.2659037.

[AN01] S. All and I. Nourbakhsh. "Insect Telepresence: Using Robotic Tele-Embodiment to Bring Insects Face-to-Face with Humans". English. In: *Autonomous Robots* 10.2 (2001), pp. 149–161. ISSN: 0929-5593. DOI: 10.1023/A:1008929919202. URL: http://dx.doi.org/10.1023/A%3A1008929919202.

[Ana+06] V. Anagnostopoulos, S. Chatzis, C. Lalos, A. Doulamis, D. Kosmopoulos, T. Varvarigou, H. Neuschmied, G. Thallinger, S. Middleton, M. Addis, E. Bustos, and F. Giorgini. "A Cross Media Platform for Personalized Leisure Entertainment: The POLYMNIA Approach". In: *Automated Production of Cross Media Content for Multi-Channel Distribution, 2006. AXMEDIS '06. Second International Conference on*. 2006, pp. 283–290. DOI: 10.1109/AXMEDIS.2006.3.

[And+10] E. F. Anderson, L. McLoughlin, F. Liarokapis, C. Peters, P. Petridis, and S. de Freitas. "Developing Serious Games for Cultural Heritage: A State-of-the-art Review". In: *Virtual Real.* 14.4 (Dec. 2010), pp. 255–275. ISSN: 1359-4338. DOI: 10.1007/s10055-010-0177-3. URL: http://dx.doi.org/10.1007/s10055-010-0177-3.

[Ard] Arduino. *Arduino (last access June 2015)*. URL: http://www.arduino.cc.

[Art] E. Arts. *The Sims 3: Pets Official Page (last access June 2015)*. URL: http://www.ea.com/the-sims-3-pets.

[Azu+01] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre. "Recent advances in augmented reality". In: *Computer Graphics and Applications, IEEE* 21.6 (2001), pp. 34–47. ISSN: 0272-1716. DOI: 10.1109/38.963459.

[Azu+97] R. T. Azuma et al. "A survey of augmented reality". In: *Presence, MIT Press* 6.4 (1997), pp. 355–385.

[Bai+05] L. Baillie, H. Kunczier, and H. Anegg. "Rolling, Rotating and Imagining in a Virtual Mobile World". In: *Proceedings of the 7th International Conference on Human Computer Interaction with Mobile Devices &Amp; Services*. MobileHCI '05. Salzburg, Austria: ACM, 2005, pp. 283–286. ISBN: 1-59593-089-2. DOI: 10.1145/1085777.1085833. URL: http://doi.acm.org/10.1145/1085777.1085833.

[Bal+07] R. K. Balan, D. Gergle, M. Satyanarayanan, and J. Herbsleb. "Simplifying Cyber Foraging for Mobile Devices". In: *Proceedings of the 5th International Conference on Mobile Systems, Applications and Services*. MobiSys '07. San Juan, Puerto Rico: ACM, 2007, pp. 272–285. ISBN: 978-1-59593-614-1. DOI: 10.1145/1247660.1247692. URL: http://doi.acm.org/10.1145/1247660.1247692.

[Bana] Bandai. *Digimon Official Page (last access April 2015)*. URL: http://www.bandai-asia.com/newsite/products.php?id=DIGIMON.

[Banb] Bandai. *Tamagotchi Official Page (last access April 2015)*. URL: http://www.bandai-asia.com/newsite/products.php?id=TAMAGOTCHI.

[Bar03] R. Bartle. *Designing Virtual Worlds*. New Riders, ISBN:0-13-101816-7, 2003.

[Bea+07] R. M. Beales, S. E. Middleton, and M. J. Addis. "Ambient multi-camera personal documentaries". In: *Visual Media Production, 2007. IETCVMP. 4th European Conference on* (2007), pp. 1–1. URL: http://eprints.soton.ac.uk/266330/.

[Bel+03] F Bellotti, R Berta, A. D. Gloria, and M Margarone. "MADE: developing edutainment applications on mobile computers". In: *Computers and Graphics* 27.4 (2003), pp. 617 –634. ISSN: 0097-8493. DOI: http://dx.doi.org/10.1016/S0097-8493(03)00105-5. URL: http://www.sciencedirect.com/science/article/pii/S0097849303001055.

[Bel+13] F. Bellotti, R. Berta, A. De Gloria, A. D'ursi, and V. Fiore. "A Serious Game Model for Cultural Heritage". In: *J. Comput. Cult. Herit.* 5.4 (Jan. 2013), 17:1–17:27. ISSN: 1556-4673. DOI: 10.1145/2399180.2399185. URL: http://doi.acm.org/10.1145/2399180.2399185.

[Blia] Blizzard. *Diablo (last access June 2015)*. URL: `http://us.battle.net/d3/en/`.

[Blib] Blizzard. *World Of Warcraft (last access June 2015)*. URL: `http://eu.battle.net/en/int?r=wow`.

[Bob+08] M. Boberg, P. Piippo, and E. Ollila. "Designing avatars". In: *Proceedings of the 3rd international conference on Digital Interactive Media in Entertainment and Arts*. DIMEA '08. Athens, Greece: ACM, 2008, pp. 232–239. ISBN: 978-1-60558-248-1. DOI: `10.1145/1413634.1413679`. URL: `http://doi.acm.org/10.1145/1413634.1413679`.

[Bon12] B. Bongers. "The Projector As Instrument". In: *Personal Ubiquitous Comput.* 16.1 (Jan. 2012), pp. 65–75. ISSN: 1617-4909. DOI: `10.1007/s00779-011-0378-0`. URL: `http://dx.doi.org/10.1007/s00779-011-0378-0`.

[Bra+04] M. Bramberger, J. Brunner, B. Rinner, and H. Schwabach. "Real-time video analysis on an embedded smart camera for traffic surveillance". In: *Real-Time and Embedded Technology and Applications Symposium, 2004. Proceedings. RTAS 2004. 10th IEEE*. 2004, pp. 174–181. DOI: `10.1109/RTTAS.2004.1317262`.

[Bra+06] M. Bramberger, A. Doblander, A. Maier, B. Rinner, and H. Schwabach. "Distributed embedded smart cameras for surveillance applications". In: *Computer* 39.2 (2006), pp. 68–75. ISSN: 0018-9162. DOI: `10.1109/MC.2006.55`.

[CD04] G. Chanda and F. Dellaert. "Grammatical methods in computer vision: An over-
view." In: *Technical Report GIT-GVU-04-29, College of Com- puting, Georgia Institute of Technology, Atlanta, GA, USA (2004) ftp://ftp.cc.gatech.edu/pub/gvu/tr/2004/04-29.pdf* (2004).

[Cha+14] K.-E. Chang, C.-T. Chang, H.-T. Hou, Y.-T. Sung, H.-L. Chao, and C.-M. Lee. "Development and behavioral pattern analysis of a mobile guide system with augmented reality for painting appreciation instruction in an art museum". In: *Computers and Education* 71.0 (2014), pp. 185 –197. ISSN: 0360-1315. DOI: `http://dx.doi.org/10.1016/j.compedu.2013.09.022`. URL: `http://www.sciencedirect.com/science/article/pii/S0360131513002868`.

[Che+00] K. Cheverst, N. Davies, K. Mitchell, A. Friday, and C. Efstratiou. "Developing a Context-aware Electronic Tourist Guide: Some Issues and Experiences". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '00. The Hague, The Netherlands: ACM, 2000, pp. 17–24. ISBN: 1-58113-216-6. DOI: `10.1145/332040.332047`. URL: `http://doi.acm.org/10.1145/332040.332047`.

[Com+y] D. Comaniciu, V. Ramesh, and P. Meer. "Kernel-based object tracking". In: *Transactions on Pattern Analysis and Machine Intelligence* 25.5 (May 2003), pp. 564–577. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2003.1195991.

[Com] T. P. Company. *Pokemon Official Page (last access April 2015)*. URL: http://www.pokemon.com.

[Con+04] J Connell, A. W. Senior, A. Hampapur, Y.-L. Tian, L Brown, and S. Pankanti. "Detection and tracking in the IBM PeopleVision system". In: *Multimedia and Expo, 2004. ICME'04. 2004 IEEE International Conference on*. Vol. 2. IEEE. 2004, pp. 1403–1406.

[Con03] D. J. Conway. *Dancing with Dragons*. ISBN: 978-1567181654, Llewellyn Publications, 2003.

[Cor+05] N. Correia, L. Alves, H. Correia, L. Romero, C. Morgado, L. Soares, J. C. Cunha, T. Romão, A. E. Dias, and J. A. Jorge. "InStory: A System for Mobile Information Access, Storytelling and Gaming Activities in Physical Spaces". In: *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*. ACE '05. Valencia, Spain: ACM, 2005, pp. 102–109. ISBN: 1-59593-110-4. DOI: 10.1145/1178477.1178491. URL: http://doi.acm.org/10.1145/1178477.1178491.

[Cor+10] N. Correia, T. Mota, R. Nóbrega, L. Silva, and A. Almeida. "A Multi-touch Tabletop for Robust Multimedia Interaction in Museums". In: *ACM International Conference on Interactive Tabletops and Surfaces*. ITS '10. Saarbrücken, Germany: ACM, 2010, pp. 117–120. ISBN: 978-1-4503-0399-6. DOI: 10.1145/1936652.1936674. URL: http://doi.acm.org/10.1145/1936652.1936674.

[CV05] S. Cotterell and F. Vahid. "A Logic Block Enabling Logic Configuration by Non-experts in Sensor Networks". In: *CHI '05 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '05. Portland, OR, USA: ACM, 2005, pp. 1925–1928. ISBN: 1-59593-002-7. DOI: 10.1145/1056808.1057058. URL: http://doi.acm.org/10.1145/1056808.1057058.

[Tob] *Creative Histories - The Josefsplatz Experience (last access June 2015)*. URL: http://www.vrvis.at/projects/josefsplatz-experience/.

[Dey+06] A. Dey, T. Sohn, S. Streng, and J. Kodama. "iCAP: Interactive Prototyping of Context-Aware Applications". English. In: *Pervasive Computing*. Ed. by K. Fishkin, B. Schiele, P. Nixon, and A. Quigley. Vol. 3968. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2006, pp. 254–271. ISBN: 978-3-540-33894-9. DOI: 10.1007/11748625_16. URL: http://dx.doi.org/10.1007/11748625_16.

[Ege99] M. Egenhofer. “Spatial information appliances: A next generation of geographic information systems”. In: *1st Brazilian workshop on geoinformatics, Campinas, Brazil*. 1999.

[Fev+11] A. Fevgas, P. Tsompanopoulou, and P. Bozanis. “iMuse Mobile Tour: A personalized multimedia museum guide opens to groups”. In: *Computers and Communications (ISCC), 2011 IEEE Symposium on*. 2011, pp. 971–975. DOI: 10.1109/ISCC.2011.5983968.

[Fro+08] P. Frohlich, M. Baldauf, P. Reichl, and R. Tobler. “Visual Presentation Challenges for Mobile Spatial Applications: Three Case Studies”. In: *Information Visualisation, 2008. IV '08. 12th International Conference*. 2008, pp. 533–538. DOI: 10.1109/IV.2008.30.

[Fr06] P. Fröhlich, R. Simon, L. Baillie, and H. Anegg. “Comparing Conceptual Designs for Mobile Access to Geo-spatial Information”. In: *Proceedings of the 8th Conference on Human-computer Interaction with Mobile Devices and Services*. MobileHCI '06. Helsinki, Finland: ACM, 2006, pp. 109–112. ISBN: 1-59593-390-5. DOI: 10.1145/1152215.1152238. URL: http://doi.acm.org/10.1145/1152215.1152238.

[Fr09] P. Fröhlich, R. Simon, and L. Baillie. “Introduction to the special issue on Mobile Spatial Interaction”. In: *Personal Ubiquitous Comput.* 13.4 (May 2009), pp. 251–253. ISSN: 1617-4909. DOI: 10.1007/s00779-008-0208-1. URL: http://dx.doi.org/10.1007/s00779-008-0208-1.

[Gam] B. F. Games. *Fish and Co Official Page (last access April 2015)*. URL: http://www.bigfishgames.com/games/2732/fishco/.

[Ghi+09] G. Ghiani, F. Paternò, and L. Spano. “Cicero Designer: An Environment for End-User Development of Multi-Device Museum Guides”. English. In: *End-User Development*. Ed. by V. Pipek, M. Rosson, B. de Ruyter, and V. Wulf. Vol. 5435. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2009, pp. 265–274. ISBN: 978-3-642-00425-4. DOI: 10.1007/978-3-642-00427-8_15. URL: http://dx.doi.org/10.1007/978-3-642-00427-8_15.

[Acm] *Grand Challenge for ACM Multimedia 2009*. 433097. Beijing, China: ACM, 2009. ISBN: 978-1-60558-608-3.

[Gra03] J. Gray. “What Next?: A Dozen Information-technology Research Goals”. In: *J. ACM* 50.1 (Jan. 2003), pp. 41–57. ISSN: 0004-5411. DOI: 10.1145/602382.602401. URL: http://doi.acm.org/10.1145/602382.602401.

[GR09] A. Greaves and E. Rukzio. “View and Share: Supporting Co-present Viewing and Sharing of Media Using Personal Projection”. In: *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*. MobileHCI ’09. Bonn, Germany: ACM, 2009, 44:1–44:4. ISBN: 978-1-60558-281-8. DOI: 10.1145/1613858.1613914. URL: http://doi.acm.org/10.1145/1613858.1613914.

[Gri+02] R. E. Grinter, P. M. Aoki, M. H. Szymanski, J. D. Thornton, A. Woodruff, and A. Hurst. “Revisiting the Visit:: Understanding How Technology Can Shape the Museum Visit”. In: *Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work*. CSCW ’02. New Orleans, Louisiana, USA: ACM, 2002, pp. 146–155. ISBN: 1-58113-560-2. DOI: 10.1145/587078.587100. URL: http://doi.acm.org/10.1145/587078.587100.

[HB05] T. Hall and L. Bannon. “Designing Ubiquitous Computing to Enhance Children’s Interaction in Museums”. In: *Proceedings of the 2005 Conference on Interaction Design and Children*. IDC ’05. Boulder, Colorado: ACM, 2005, pp. 62–69. ISBN: 1-59593-096-5. DOI: 10.1145/1109540.1109549. URL: http://doi.acm.org/10.1145/1109540.1109549.

[Han+08] A. Hang, E. Rukzio, and A. Greaves. “Projector Phone: A Study of Using Mobile Phones with Integrated Projector for Interaction with Maps”. In: *Proceedings of the 10th International Conference on Human Computer Interaction with Mobile Devices and Services*. MobileHCI ’08. Amsterdam, The Netherlands: ACM, 2008, pp. 207–216. ISBN: 978-1-59593-952-4. DOI: 10.1145/1409240.1409263. URL: http://doi.acm.org/10.1145/1409240.1409263.

[Has] Hasbro. *Furby Official Page (last access April 2015)*. URL: http://www.hasbro.com/furby/en_us/.

[HT04] L. Hempel and E Topfer. “CCTV in Europe, Final Report”. In: *Final Report, The Urbaneye Working Papers Series, edited by 5th Framework Programme of the European Commission* (2004).

[Her+03] F. Hermann, G. Bieber, and A. Duesterhoeft. “Egocentric maps on mobile devices”. In: *the Proc. International Workshop on Mobile Computing*. Citeseer. IRB Verlag, 2003.

[Hor10] E. Hornecker. “Creative Idea Exploration Within the Structure of a Guiding Framework: The Card Brainstorming Game”. In: *Proceedings of the Fourth International Conference on Tangible, Embedded, and Embodied Interaction*. TEI ’10. Cambridge, Massachusetts, USA: ACM, 2010, pp. 101–108. ISBN: 978-1-60558-841-4. DOI: 10.1145/1709886.1709905. URL: http://doi.acm.org/10.1145/1709886.1709905.

[HF05] S. Hsi and H. Fait. "RFID Enhances Visitors' Museum Experience at the Exploratorium". In: *Commun. ACM* 48.9 (Sept. 2005), pp. 60–65. ISSN: 0001-0782. DOI: 10.1145/1081992.1082021. URL: http://doi.acm.org/10.1145/1081992.1082021.

[Hua+11] Y.-P. Huang, S.-S. Wang, and F. E. Sandnes. "RFID-Based Guide Gives Museum Visitors More Freedom". In: *IT Professional* 13.2 (2011), pp. 25–29. ISSN: 1520-9202. DOI: http://doi.ieeecomputersociety.org/10.1109/MITP.2011.33.

[Ipe] *IPerG, Mixed Reality Lab (last access June 2015)*. URL: http://www.pervasive-gaming.org.

[IB00] Y. Ivanov and A. Bobick. "Recognition of visual activities and interactions by stochastic parsing". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 22.8 (2000), pp. 852–872. ISSN: 0162-8828. DOI: 10.1109/34.868686.

[JC06] S.-W. Joo and R. Chellappa. "Attribute Grammar-Based Event Recognition and Anomaly Detection". In: *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW '06. Conference on*. 2006, pp. 107–107. DOI: 10.1109/CVPRW.2006.32.

[Kab13] K. Kabassi. "Personalisation Systems for Cultural Tourism". English. In: *Multimedia Services in Intelligent Environments*. Ed. by G. A. Tsihrintzis, M. Virvou, and L. C. Jain. Vol. 25. Smart Innovation, Systems and Technologies. Springer International Publishing, 2013, pp. 101–111. ISBN: 978-3-319-00374-0. DOI: 10.1007/978-3-319-00375-7_7. URL: http://dx.doi.org/10.1007/978-3-319-00375-7_7.

[Kaw+10] F. Kawsar, E. Rukzio, and G. Kortuem. "An Explorative Comparison of Magic Lens and Personal Projection for Interacting with Smart Objects". In: *Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services*. MobileHCI '10. Lisbon, Portugal: ACM, 2010, pp. 157–160. ISBN: 978-1-60558-835-3. DOI: 10.1145/1851600.1851627. URL: http://doi.acm.org/10.1145/1851600.1851627.

[Kru+12] D. M. Krum, E. A. Suma, and M. Bolas. "Augmented Reality Using Personal Projection and Retroreflection". In: *Personal Ubiquitous Comput.* 16.1 (Jan. 2012), pp. 17–26. ISSN: 1617-4909. DOI: 10.1007/s00779-011-0374-4. URL: http://dx.doi.org/10.1007/s00779-011-0374-4.

[Kue+14] H. Kuehne, A. Arslan, and T. Serre. "The Language of Actions: Recovering the Syntax and Semantics of Goal-Directed Human Activities". In: *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. 2014, pp. 780–787. DOI: 10.1109/CVPR.2014.105.

[Kuf+11] T. Kuflik, O. Stock, M. Zancanaro, A. Gorfinkel, S. Jbara, S. Kats, J. Sheidin, and N. Kashtan. “A Visitor’s Guide in an Active Museum: Presentations, Communications, and Reflection”. In: *J. Comput. Cult. Herit.* 3.3 (Feb. 2011), 11:1–11:25. ISSN: 1556-4673. DOI: 10.1145/1921614.1921618. URL: http://doi.acm.org/10.1145/1921614.1921618.

[Lan03] G. Lane. “Urban Tapestries: Wireless Networking, Public Authoring and Social Knowledge”. In: *Personal Ubiquitous Comput.* 7.3-4 (July 2003), pp. 169–175. ISSN: 1617-4909. DOI: 10.1007/s00779-003-0229-8. URL: http://dx.doi.org/10.1007/s00779-003-0229-8.

[Lan+13] J. Lanir, T. Kuflik, E. Dim, A. J. Wecker, and O. Stock. “The Influence of a Location-Aware Mobile Guide on Museum Visitors’ Behavior”. In: *Interacting with Computers* 25.6 (2013), pp. 443–460. DOI: 10.1093/iwc/iwt002. URL: http://dx.doi.org/10.1093/iwc/iwt002.

[LP04] Y. Laurillau and F. Paternò. “Supporting Museum Co-visits Using Mobile Devices”. English. In: *Mobile Human-Computer Interaction - MobileHCI 2004*. Ed. by S. Brewster and M. Dunlop. Vol. 3160. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2004, pp. 451–455. ISBN: 978-3-540-23086-1. DOI: 10.1007/978-3-540-28637-0_55. URL: http://dx.doi.org/10.1007/978-3-540-28637-0_55.

[Leg] Lego. *Lego Mindstorms (last access June 2015)*. URL: http://mindstorms.lego.com.

[Leh+07] D. vom Lehn, J. Hindmarsh, P. Luff, and C. Heath. “Engaging Constable: Revealing Art with New Technology”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’07. San Jose, California, USA: ACM, 2007, pp. 1485–1494. ISBN: 978-1-59593-593-9. DOI: 10.1145/1240624.1240848. URL: http://doi.acm.org/10.1145/1240624.1240848.

[Leu+11] M. Leung, M. Tomitsch, and A. Vande Moere. “Designing a Personal Visualization Projection of Online Social Identity”. In: *CHI ’11 Extended Abstracts on Human Factors in Computing Systems*. CHI EA ’11. Vancouver, BC, Canada: ACM, 2011, pp. 1843–1848. ISBN: 978-1-4503-0268-5. DOI: 10.1145/1979742.1979882. URL: http://doi.acm.org/10.1145/1979742.1979882.

[Li+14] Y. Li, Z. Wu, S. Karanam, and R. J. Radke. “Real-World Re-Identification in an Airport Camera Network”. In: *Proceedings of the International Conference on Distributed Smart Cameras*. ICDSC ’14. Venezia Mestre, Italy: ACM, 2014, 35:1–35:6. ISBN: 978-1-4503-2925-5. DOI: 10.1145/2659021.2659039. URL: http://doi.acm.org/10.1145/2659021.2659039.

[Lis+14] G. Lisanti, I. Masi, and A. Del Bimbo. "Matching People Across Camera Views Using Kernel Canonical Correlation Analysis". In: *Proceedings of the International Conference on Distributed Smart Cameras*. ICDSC '14. Venezia Mestre, Italy: ACM, 2014, 10:1–10:6. ISBN: 978-1-4503-2925-5. DOI: 10.1145/2659021.2659036. URL: http://doi.acm.org/10.1145/2659021.2659036.

[Loh] S. Lohr. *Museums Morph Digitally-The Met and Other Museums Adapt to the Digital Age (last access July 2015)*. URL: http://www.nytimes.com/2014/10/26/arts/artsspecial/the-met-and-other-museums-adapt-to-the-digital-age.html?_r=1.

[Lou] "Louvre Museum". In: *(online: January 2012) http://www.louvre.fr* (2009).

[Man09] S. Mann. "Wearable Wireless Webcam and Telemetry". In: *(online July 2015) http://wearcam.org/netcam.html* (2009).

[Mar+08] T. Martins, T. Romão, C. Sommerer, L. Mignonneau, and N. Correia. "Towards an Interface for Untethered Ubiquitous Gaming". In: *Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology*. ACE '08. Yokohama, Japan: ACM, 2008, pp. 26–33. ISBN: 978-1-60558-393-8. DOI: 10.1145/1501750.1501757. URL: http://doi.acm.org/10.1145/1501750.1501757.

[MV06] J. Mattila and A. Väätänen. "UbiPlay: An Interactive Playground and Visual Programming Tools for Children". In: *Proceedings of the 2006 Conference on Interaction Design and Children*. IDC '06. Tampere, Finland: ACM, 2006, pp. 129–136. DOI: 10.1145/1139073.1139106. URL: http://doi.acm.org/10.1145/1139073.1139106.

[Mica] Microsoft. *Kinect Animals Official Page (last access April 2015)*. URL: http://www.xbox.com/en-US/marketplace/SplashPages/kinectimals.

[Micb] Microsoft. *Kinect (last access July 2015)*. URL: https://www.microsoft.com/en-us/kinectforwindows/.

[Micc] Microsoft. *Kodu (last access June 2015)*. URL: http://research.microsoft.com/en-us/projects/kodu/.

[Micd] Microsoft. *Microsoft Visual Basic (last access June 2015)*. URL: https://msdn.microsoft.com/en-us/library/aa227231(v=vs.60).aspx.

[Mice] Microvision. *Official website (last access June 2015)*. URL: http://www.microvision.com.

[Mil+95] P. Milgram, H. Takemura, A. Utsumi, and F. Kishino. *Augmented reality: a class of displays on the reality-virtuality continuum*. 1995. DOI: 10.1117/12.197321. URL: http://dx.doi.org/10.1117/12.197321.

[Min+03] D. Minnen, I. Essa, and T. Starner. "Expectation grammars: leveraging high-level expectations for activity recognition". In: *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*. Vol. 2. 2003, II–626–II–632 vol.2. DOI: 10.1109/CVPR.2003.1211525.

[MM09] P. Mistry and P. Maes. "SixthSense: A Wearable Gestural Interface". In: *ACM SIGGRAPH ASIA 2009 Sketches*. SIGGRAPH ASIA '09. Yokohama, Japan: ACM, 2009, 11:1–11:1. DOI: 10.1145/1667146.1667160. URL: http://doi.acm.org/10.1145/1667146.1667160.

[Mis+09] P. Mistry, P. Maes, and L. Chang. "WUW - Wear Ur World: A Wearable Gestural Interface". In: *CHI '09 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '09. Boston, MA, USA: ACM, 2009, pp. 4111–4116. ISBN: 978-1-60558-247-4. DOI: 10.1145/1520340.1520626. URL: http://doi.acm.org/10.1145/1520340.1520626.

[MIT] MIT. *Scratch Programming Language (last access June 2015)*. URL: https://scratch.mit.edu.

[Mod+09] A. Mody, M. Akram, K. Rony, M. Aman, and R. Kamoua. "Enhancing user experience at museums using smart phones with RFID". In: *Systems, Applications and Technology Conference, 2009. LISAT '09. IEEE Long Island*. 2009, pp. 1–5. DOI: 10.1109/LISAT.2009.5031573.

[Moj] Mojang. *Minecraft (last access June 2015)*. URL: https://minecraft.net.

[ME02] D. Moore and I. Essa. "Recognizing Multitasked Activities from Video Using Stochastic Context-free Grammar". In: *Eighteenth National Conference on Artificial Intelligence*. Edmonton, Alberta, Canada: American Association for Artificial Intelligence, 2002, pp. 770–776. ISBN: 0-262-51129-0. URL: http://dl.acm.org/citation.cfm?id=777092.777211.

[Mor+14] M. Mortara, C. E. Catalano, F. Bellotti, G. Fiucci, M. Houry-Panchetti, and P. Petridis. "Learning cultural heritage by serious games". In: *Journal of Cultural Heritage* 15.3 (2014), pp. 318 –325. ISSN: 1296-2074. DOI: http://dx.doi.org/10.1016/j.culher.2013.04.004. URL: http://www.sciencedirect.com/science/article/pii/S1296207413001349.

[Mul+12] A. Mulloni, H. Seichter, and D. Schmalstieg. "Indoor Navigation with Mixed Reality World-in-miniature Views and Sparse Localization on Mobile Devices". In: *Proceedings of the International Working Conference on Advanced Visual Interfaces*. AVI '12. Capri Island, Italy: ACM, 2012, pp. 212–215. ISBN: 978-1-4503-1287-5. DOI: 10.1145/2254556.2254595. URL: http://doi.acm.org/10.1145/2254556.2254595.

[ML] L. Museum and S. Life. *The Second Louvre Museum (last access June 2015)*. URL: secondlife.com/.

[Nin] Nintendo. *Nintendogs + cats Official Page (last access June 2015)*. URL: http://nintendogspluscats.nintendo.com.

[Nob11] R. Nobrega. "Modeling Places for Interactive Media and Entertainment Applications". In: *CHI '11 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '11. Vancouver, BC, Canada: ACM, 2011, pp. 1081–1084. ISBN: 978-1-4503-0268-5. DOI: 10.1145/1979742.1979692. URL: http://doi.acm.org/10.1145/1979742.1979692.

[NC13] R. Nóbrega and N. Correia. "Dynamic Insertion of Virtual Objects in Photographs". In: *International Journal of Creative Interfaces and Computer Graphics (IJCICG)* 4.2 (2013), pp. 22–39.

[Nor02] D. A. Norman. *The Design of Everyday Things*. 2002. ISBN: 0465067107. URL: http://www.worldcat.org/isbn/0465067107.

[Ohs+03] Y. Ohshima, J. Maloney, and A. Ogden. "The Parks PDA: A Handheld Device for Theme Park Guests in Squeak". In: *Companion of the 18th Annual ACM SIGPLAN Conference on Object-oriented Programming, Systems, Languages, and Applications*. OOPSLA '03. Anaheim, CA, USA: ACM, 2003, pp. 370–380. ISBN: 1-58113-751-6. DOI: 10.1145/949344.949444. URL: http://doi.acm.org/10.1145/949344.949444.

[Ong+05] S. K. Ong, M. L. Yuan, and A. Y. C. Nee. "Tracking Points Using Projective Reconstruction for Augmented Reality". In: *Proceedings of the 3rd International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia*. GRAPHITE '05. Dunedin, New Zealand: ACM, 2005, pp. 421–424. ISBN: 1-59593-201-1. DOI: 10.1145/1101389.1101472. URL: http://doi.acm.org/10.1145/1101389.1101472.

[OR05] S. C. W. Ong and S. Ranganath. "Automatic Sign Language Analysis: A Survey and the Future Beyond Lexical Meaning". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 27.6 (June 2005), pp. 873–891. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2005.112. URL: http://dx.doi.org/10.1109/TPAMI.2005.112.

[OS00] R. Oppermann and M. Specht. "A Context-Sensitive Nomadic Exhibition Guide". In: *Proceedings of the 2Nd International Symposium on Handheld and Ubiquitous Computing*. HUC '00. Bristol, UK: Springer-Verlag, 2000, pp. 127–142. ISBN: 3-540-41093-7. URL: http://dl.acm.org/citation.cfm?id=647986.741314.

[Ou+04] S. Ou, D. Karuppiah, A. Fagg, and E. Riseman. "An augmented virtual reality interface for assistive monitoring of smart spaces". In: *Pervasive Computing and Communications, 2004. PerCom 2004. Proceedings of the Second IEEE Annual Conference on*. 2004, pp. 33–42. DOI: 10.1109/PERCOM.2004.1276843.

[Oul+09] A. Oulasvirta, S. Estlander, and A. Nurminen. “Embodied Interaction with a 3D Versus 2D Mobile Map”. In: *Personal Ubiquitous Comput.* 13.4 (May 2009), pp. 303–320. ISSN: 1617-4909. DOI: 10.1007/s00779-008-0209-0. URL: http://dx.doi.org/10.1007/s00779-008-0209-0.

[Ped+13] R. W. D. Pedro, F. L. S. Nunes, and A. Machado-Lima. “Using Grammars for Pattern Recognition in Images: A Systematic Review”. In: *ACM Comput. Surv.* 46.2 (Nov. 2013), 26:1–26:34. ISSN: 0360-0300. DOI: 10.1145/2543581.2543593. URL: http://doi.acm.org/10.1145/2543581.2543593.

[Per+03] P. Persson, F. Espinoza, P. Fagerberg, A. Sandin, and R. Cöster. “GeoNotes: A Location-Based Information System for Public Spaces”. English. In: *Designing Information Spaces: The Social Navigation Approach*. Ed. by K. Höök, D. Benyon, and A. Munro. Computer Supported Cooperative Work. Springer London, 2003, pp. 151–173. ISBN: 978-1-85233-661-5. DOI: 10.1007/978-1-4471-0035-5_7. URL: http://dx.doi.org/10.1007/978-1-4471-0035-5_7.

[Pha+04] T. Phan, G. Zorpas, and R. Bagrodia. “Middleware support for reconciling client updates and data transcoding”. In: *MobiSys ’04: Proceedings of the 2nd international conference on Mobile systems, applications, and services*. Boston, MA, USA: ACM, 2004, pp. 139–152. ISBN: 1-58113-793-1. DOI: http://doi.acm.org/10.1145/990064.990083.

[PA14] E. Pietroni and A. Adami. “Interacting with Virtual Reconstructions in Museums: The Etruscanning Project”. In: *J. Comput. Cult. Herit.* 7.2 (June 2014), 9:1–9:29. ISSN: 1556-4673. DOI: 10.1145/2611375. URL: http://doi.acm.org/10.1145/2611375.

[Pri+02] S. Prince, A. D. Cheok, F. Farbiz, T. Williamson, N. Johnson, M. Billinghurst, and H. Kato. “3-D Live: Real Time Interaction for Mixed Reality”. In: *Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work*. CSCW ’02. New Orleans, Louisiana, USA: ACM, 2002, pp. 364–371. ISBN: 1-58113-560-2. DOI: 10.1145/587078.587129. URL: http://doi.acm.org/10.1145/587078.587129.

[Ras+06] N. T. Rasmussen, M. Störring, T. B. Moeslund, and E. Granum. “Real-time tracking for virtual environments using SCAAT Kalman filtering and unsynchronized cameras”. In: *International Conference on Computer Vision Theory and Applications (VISAPP)(February 2006)*. Citeseer. 2006.

[RK13] O. Raztne and V. Kaptelinin. “Design Choices and Museum Experience: A Design-Based Study of a Mobile Museum App”. English. In: *Human Computer Interaction*. Ed. by C. Collazos, A. Liborio, and C. Rusu. Vol. 8278. Lecture Notes in Computer Science. Springer International Publishing, 2013, pp. 9–13.

ISBN: 978-3-319-03067-8. DOI: 10.1007/978-3-319-03068-5_3. URL: http://dx.doi.org/10.1007/978-3-319-03068-5_3.

[RC11] S. Reis and N. Correia. "An Imaginary Friend That Connects with the User's Emotions". In: *Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology*. ACE '11. Lisbon, Portugal: ACM, 2011, 1:1–1:8. ISBN: 978-1-4503-0827-4. DOI: 10.1145/2071423.2071425. URL: http://doi.acm.org/10.1145/2071423.2071425.

[RN95] J. Rekimoto and K. Nagao. "The World Through the Computer: Computer Augmented Interaction with Real World Environments". In: *Proceedings of the 8th Annual ACM Symposium on User Interface and Software Technology*. UIST '95. Pittsburgh, Pennsylvania, USA: ACM, 1995, pp. 29–36. ISBN: 0-89791-709-X. DOI: 10.1145/215585.215639. URL: http://doi.acm.org/10.1145/215585.215639.

[Rin+07] B. Rinner, M. Jovanovic, and M. Quaritsch. "Embedded Middleware on Distributed Smart Cameras". In: *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on* 4 (2007), pp. IV–1381–IV–1384. ISSN: 1520-6149. DOI: 10.1109/ICASSP.2007.367336.

[Roj+14] S. L. Rojas, L. Oppermann, L. Blum, and M. Wolpers. "Natural Europe Educational Games Suite: Using Structured Museum-data for Creating Mobile Educational Games". In: *Proceedings of the 11th Conference on Advances in Computer Entertainment Technology*. ACE '14. Funchal, Portugal: ACM, 2014, 6:1–6:6. ISBN: 978-1-4503-2945-3. DOI: 10.1145/2663806.2663841. URL: http://doi.acm.org/10.1145/2663806.2663841.

[Rom+04] T. Romao, N. Correia, E. Dias, J. Danado, A. Trabuco, C. Santos, R. Santos, A. Camara, and E. Nobre. "ANTS-Augmented Environments". In: *Computers &Amp; Graphics* 28.5 (2004), pp. 625 –633. ISSN: 0097-8493. DOI: http://dx.doi.org/10.1016/j.cag.2004.06.001. URL: http://www.sciencedirect.com/science/article/pii/S0097849304000858.

[Rou+01] M. Roussou, P. Trahanias, G. Giannoulis, G. Kamarinos, A. Argyros, D. Tsakiris, P. Georgiadis, W. Burgard, D. Haehnel, A. Cremers, D. Schulz, M. Moors, E. Spirtounias, M. Marianthi, V. Savvaides, A. Reitelman, D. Konstantios, and A. Katselaki. "Experiences from the Use of a Robotic Avatar in a Museum Setting". In: *Proceedings of the 2001 Conference on Virtual Reality, Archeology, and Cultural Heritage*. VAST '01. Glyfada, Greece: ACM, 2001, pp. 153–160. ISBN: 1-58113-447-9. DOI: 10.1145/584993.585017. URL: http://doi.acm.org/10.1145/584993.585017.

[Rub+15] I. Rubino, C. Barberis, J. Xhembulla, and G. Malnati. "Integrating a Location-Based Mobile Game in the Museum Visit: Evaluating Visitors&Rsquo; Behaviour and Learning". In: *J. Comput. Cult. Herit.* 8.3 (May 2015), 15:1–15:18.

ISSN: 1556-4673. DOI: 10.1145/2724723. URL: http://doi.acm.org/10.1145/2724723.

[RA06] M. Ryoo and J. Aggarwal. "Recognition of Composite Human Activities through Context-Free Grammar Based Representation". In: *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*. Vol. 2. 2006, pp. 1709–1718. DOI: 10.1109/CVPR.2006.242.

[Sam] Samnsung. *Samsung Beam Mobile Device (last access June 2015)*. URL: http://www.samsung.com/uk/consumer/mobile-devices/smartphones/android/GT-I8530BAABTU.

[SC10] R. Santos and N. Correia. "vuSpot: An Infrastructure for Augmented Information Sharing Using Surveillance Cameras". In: *Proceedings of the Fourth ACM/IEEE International Conference on Distributed Smart Cameras*. ICDSC '10. Atlanta, Georgia: ACM, 2010, pp. 230–231. ISBN: 978-1-4503-0317-0. DOI: 10.1145/1865987.1866025. URL: http://doi.acm.org/10.1145/1865987.1866025.

[SC12] R. Santos and N. Correia. "UBI, the Guardian Dragon: Your Virtual Sidekick". In: *Proceedings of the 9th International Conference on Advances in Computer Entertainment*. ACE'12. Kathmandu, Nepal: Springer-Verlag, 2012, pp. 368–383. ISBN: 978-3-642-34291-2. DOI: 10.1007/978-3-642-34292-9_26. URL: http://dx.doi.org/10.1007/978-3-642-34292-9_26.

[Saw+02] H. S. Sawhney, A. Arpa, R. Kumar, S. Samarasekera, M. Aggarwal, S. Hsu, D. Nister, and K. Hanna. "Video Flashlights: Real Time Rendering of Multiple Videos for Immersive Model Visualization". In: *Proceedings of the 13th Eurographics Workshop on Rendering*. EGRW '02. Pisa, Italy: Eurographics Association, 2002, pp. 157–168. ISBN: 1-58113-534-3. URL: http://dl.acm.org/citation.cfm?id=581896.581918.

[Sch+09a] G. Schall, E. Mendez, E. Kruijff, E. Veas, S. Junghanns, B. Reitinger, and D. Schmalstieg. "Handheld Augmented Reality for underground infrastructure visualization". English. In: *Personal and Ubiquitous Computing* 13.4 (2009), pp. 281–291. ISSN: 1617-4909. DOI: 10.1007/s00779-008-0204-5. URL: http://dx.doi.org/10.1007/s00779-008-0204-5.

[SW07] D. Schmalstieg and D. Wagner. "Experiences with Handheld Augmented Reality". In: *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*. 2007, pp. 3–18. DOI: 10.1109/ISMAR.2007.4538819.

[Sch+09b] J. Schöning, M. Rohs, S. Kratz, M. Löchtefeld, and A. Krüger. "Map Torchlight: A Mobile Augmented Reality Camera Projector Unit". In: *CHI '09 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '09. Boston, MA, USA: ACM, 2009, pp. 3841–3846. ISBN: 978-1-60558-247-4. DOI: 10.1145/

1520340.1520581. URL: http://doi.acm.org/10.1145/1520340.1520581.

[Sen+14] S. Sen, S. Roy, and S. Sarkar. "A Proposal for Enhancing Museum Visiting Experience Implementing Active RFID Technology". In: *Advances in Computing and Communications (ICACC), 2014 Fourth International Conference on*. 2014, pp. 295–298. DOI: 10.1109/ICACC.2014.76.

[Sim+06] R. Simon, P. Fröhlich, and H. Anegg. "Beyond Location Based – The Spatially Aware Mobile Phone". English. In: *Web and Wireless Geographical Information Systems*. Ed. by J. Carswell and T. Tezuka. Vol. 4295. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2006, pp. 12–21. ISBN: 978-3-540-49466-9. DOI: 10.1007/11935148_2. URL: http://dx.doi.org/10.1007/11935148_2.

[Sof] M. Software. *Visuino: visual development for arduino (last access June 2015)*. URL: www.visuino.com.

[Son+11] Y. Song, D. Demirdjian, and R. Davis. "Tracking body and hands for gesture recognition: NATOPS aircraft handling signals database". In: *Automatic Face Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*. 2011, pp. 500–506. DOI: 10.1109/FG.2011.5771448.

[Son+12] Y. Song, D. Demirdjian, and R. Davis. "Continuous Body and Hand Gesture Recognition for Natural Human-computer Interaction". In: *ACM Trans. Interact. Intell. Syst.* 2.1 (Mar. 2012), 5:1–5:28. ISSN: 2160-6455. DOI: 10.1145/2133366.2133371. URL: http://doi.acm.org/10.1145/2133366.2133371.

[Spa04] F. Sparacino. "Scenographies of the Past and Museums of the Future: From the Wunderkammer to Body-driven Interactive Narrative Spaces". In: *Proceedings of the 12th Annual ACM International Conference on Multimedia*. MULTIMEDIA '04. New York, NY, USA: ACM, 2004, pp. 72–79. ISBN: 1-58113-893-8. DOI: 10.1145/1027527.1027541. URL: http://doi.acm.org/10.1145/1027527.1027541.

[TP12] Y. Takeuchi and K. Perlin. "ClayVision: The (Elastic) Image of the City". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '12. Austin, Texas, USA: ACM, 2012, pp. 2411–2420. ISBN: 978-1-4503-1015-4. DOI: 10.1145/2207676.2208404. URL: http://doi.acm.org/10.1145/2207676.2208404.

[TI+15] E. M. Taranta II, T. K. Simons, R. Sukthankar, and J. J. Laviola Jr. "Exploring the Benefits of Context in 3D Gesture Recognition for Game-Based Virtual Environments". In: *ACM Trans. Interact. Intell. Syst.* 5.1 (Mar. 2015), 1:1–1:34. ISSN: 2160-6455. DOI: 10.1145/2656345. URL: http://doi.acm.org/10.1145/2656345.

[Tes+08] R. Tesoriero, J. Gallud, M. Lozano, and V. Penichet. "Using active and passive RFID technology to support indoor location-aware systems". In: *Consumer Electronics, IEEE Transactions on* 54.2 (2008), pp. 578–583. ISSN: 0098-3063. DOI: 10.1109/TCE.2008.4560133.

[Tra+00] P. Trahanias, A. Argyros, D. Tsakiris, A. Cremers, D. Schulz, W. Burgard, D. Haehnel, V. Savvaides, G. Giannoulis, M. Coliou, Y Coliou, G. Kamarinos, P. Friess, D. Konstantios, and A. Katselaki. "TOURBOT - Interactive Museum Tele-presence Through Robotic Atavars - Project Presentation and Prospects". In: *Proc. of the 9th International World Wide Web Conference, Amsterdam, Netherlands* 7 (2000).

[Val98] J. R. Vallino. "Interactive augmented reality". PhD thesis. University of Rochester, 1998.

[Vez+08] R. Vezzani, S. Calderara, P. Piccinini, and R. Cucchiara. "Smoke Detection in Video Surveillance: The Use of ViSOR (Video Surveillance On-line Repository)". In: *Proceedings of the 2008 International Conference on Content-based Image and Video Retrieval*. CIVR '08. Niagara Falls, Canada: ACM, 2008, pp. 289–298. ISBN: 978-1-60558-070-8. DOI: 10.1145/1386352.1386392. URL: http://doi.acm.org/10.1145/1386352.1386392.

[Vez+13] R. Vezzani, D. Baltieri, and R. Cucchiara. "People Reidentification in Surveillance and Forensics: A Survey". In: *ACM Comput. Surv.* 46.2 (Dec. 2013), 29:1–29:37. ISSN: 0360-0300. DOI: 10.1145/2543581.2543596. URL: http://doi.acm.org/10.1145/2543581.2543596.

[Vio+04] P. Viola, M. J. Jones, and M. J. Jones. "Robust Real-Time Face Detection". In: *Int. J. Comput. Vision* 57.2 (2004), pp. 137–154. ISSN: 0920-5691. DOI: http://dx.doi.org/10.1023/B:VISI.0000013087.49260.fb.

[Vis] J. Visser. *Museums in Time of Social and Technological Change (last access June 2015)*. URL: http://themuseumofthefuture.com/2014/04/18/museums-in-times-of-social-and-technological-change/.

[Wag03] D. Wagner D.; Schmalstieg. "First steps towards handheld augmented reality". In: *Wearable Computers, 2003. Proceedings. Seventh IEEE International Symposium on* (2003), pp. 127–135.

[Wag+05] D. Wagner, T. Pintaric, F. Ledermann, and D. Schmalstieg. "Towards Massively Multi-user Augmented Reality on Handheld Devices". English. In: *Pervasive Computing*. Ed. by H.-W. Gellersen, R. Want, and A. Schmidt. Vol. 3468. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2005, pp. 208–219. ISBN: 978-3-540-26008-0. DOI: 10.1007/11428572_13. URL: http://dx.doi.org/10.1007/11428572_13.

[Wag+06] D. Wagner, M. Billinghurst, and D. Schmalstieg. “How real should virtual characters be?” In: *Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology*. ACE ’06. Hollywood, California: ACM, 2006. ISBN: 1-59593-380-8. DOI: 10.1145/1178823.1178891. URL: http://doi.acm.org/10.1145/1178823.1178891.

[Wag+10] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg. “Real-Time Detection and Tracking for Augmented Reality on Mobile Phones”. In: *Visualization and Computer Graphics, IEEE Transactions on* 16.3 (2010), pp. 355–368. ISSN: 1077-2626. DOI: 10.1109/TVCG.2009.99.

[Wei+14] W. Wei, H. Ma, H. Zhang, Y. Gao, and Z. Wang. “Person Re-identification Based on Human Body Parts Signature”. In: *Proceedings of the International Conference on Distributed Smart Cameras*. ICDSC ’14. Venezia Mestre, Italy: ACM, 2014, 9:1–9:6. ISBN: 978-1-4503-2925-5. DOI: 10.1145/2659021.2659035. URL: http://doi.acm.org/10.1145/2659021.2659035.

[Wei14] L. Wein. “Visual Recognition in Museum Guide Apps: Do Visitors Want It?” In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’14. Toronto, Ontario, Canada: ACM, 2014, pp. 635–638. ISBN: 978-1-4503-2473-1. DOI: 10.1145/2556288.2557270. URL: http://doi.acm.org/10.1145/2556288.2557270.

[Wil+13] K. D. D. Willis, T. Shiratori, and M. Mahler. “HideOut: Mobile Projector Interaction with Tangible Objects and Surfaces”. In: *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction*. TEI ’13. Barcelona, Spain: ACM, 2013, pp. 331–338. ISBN: 978-1-4503-1898-3. DOI: 10.1145/2460625.2460682. URL: http://doi.acm.org/10.1145/2460625.2460682.

[Wil+11] K. D. Willis, I. Poupyrev, S. E. Hudson, and M. Mahler. “SideBySide: Ad-hoc Multi-user Interaction with Handheld Projectors”. In: *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*. UIST ’11. Santa Barbara, California, USA: ACM, 2011, pp. 431–440. ISBN: 978-1-4503-0716-1. DOI: 10.1145/2047196.2047254. URL: http://doi.acm.org/10.1145/2047196.2047254.

[Wil+12] M. L. Wilson, D. Craggs, S. Robinson, M. Jones, and K. Brimble. “Pico-ing into the Future of Mobile Projection and Contexts”. In: *Personal Ubiquitous Comput.* 16.1 (Jan. 2012), pp. 39–52. ISSN: 1617-4909. DOI: 10.1007/s00779-011-0376-2. URL: http://dx.doi.org/10.1007/s00779-011-0376-2.

[Win+11] C. Winkler, C. Reinartz, D. Nowacka, and E. Rukzio. “Interactive Phone Call: Synchronous Remote Collaboration and Projected Interactive Surfaces”. In: *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*. ITS ’11. Kobe, Japan: ACM, 2011, pp. 61–70. ISBN: 978-1-4503-0871-7.

DOI: 10.1145/2076354.2076367. URL: http://doi.acm.org/10.1145/2076354.2076367.

[Win+14] C. Winkler, J. Seifert, D. Dobbelstein, and E. Rukzio. "Pervasive Information Through Constant Personal Projection: The Ambient Mobile Pervasive Display (AMP-D)". In: *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*. CHI '14. Toronto, Ontario, Canada: ACM, 2014, pp. 4117–4126. ISBN: 978-1-4503-2473-1. DOI: 10.1145/2556288.2557365. URL: http://doi.acm.org/10.1145/2556288.2557365.

[Woo+04] E. Woods, M. Billinghurst, J. Looser, G. Aldridge, D. Brown, B. Garrie, and C. Nelles. "Augmenting the Science Centre and Museum Experience". In: *Proceedings of the 2Nd International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia*. GRAPHITE '04. Singapore: ACM, 2004, pp. 230–236. ISBN: 1-58113-883-0. DOI: 10.1145/988834.988873. URL: http://doi.acm.org/10.1145/988834.988873.

[Wre+97] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland. "Pfinder: real-time tracking of the human body". In: *Transactions on Pattern Analysis and Machine Intelligence* 19.7 (1997), pp. 780–785. ISSN: 0162-8828. DOI: 10.1109/34.598236.

[Yan+08] K. Yang, S. Ou, and H.-H. Chen. "On effective offloading services for resource-constrained mobile devices running heavier mobile internet applications". In: *Communications Magazine, IEEE* 46.1 (2008), pp. 56–63. ISSN: 0163-6804. DOI: 10.1109/MCOM.2008.4427231.

[Zaf+11] Z. Zafrulla, H. Brashear, T. Starner, H. Hamilton, and P. Presti. "American Sign Language Recognition with the Kinect". In: *Proceedings of the 13th International Conference on Multimodal Interfaces*. ICMI '11. Alicante, Spain: ACM, 2011, pp. 279–286. ISBN: 978-1-4503-0641-6. DOI: 10.1145/2070481.2070532. URL: http://doi.acm.org/10.1145/2070481.2070532.

[Zha+14] Z. Zhang, H. Ma, H. Fu, and H. Huang. "Distributed Human Action Recognition via 2D Conditional Random Field". In: *Proceedings of the International Conference on Distributed Smart Cameras*. ICDSC '14. Venezia Mestre, Italy: ACM, 2014, 4:1–4:6. ISBN: 978-1-4503-2925-5. DOI: 10.1145/2659021.2659030. URL: http://doi.acm.org/10.1145/2659021.2659030.

[Zha+03a] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. "Face Recognition: A Literature Survey". In: *ACM Comput. Surv.* 35.4 (Dec. 2003), pp. 399–458. ISSN: 0360-0300. DOI: 10.1145/954339.954342. URL: http://doi.acm.org/10.1145/954339.954342.

[Zha+03b] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. "Face recognition: A literature survey". In: *ACM Comput. Surv.* 35.4 (2003), pp. 399–458. ISSN: 0360-0300. DOI: http://doi.acm.org/10.1145/954339.954342.

[Zyna] Zynga. *Farmville (last access June 2015)*. URL: `https://zynga.com/games/farmville-2`.

[Zynb] Zynga. *Official website (last access April 2015)*. URL: `https://zynga.com`.

APPENDIX

A

INFRASTRUCTURE EXAMPLES

Five case studies where the vuSpot infrastructure was used are presented in the next sections. These case studies correspond to the applications developed during this research and that are described in Chapter 4. For each case study the corresponding architecture is presented.

A.1 UBI, The Guardian Dragon

Several modules of the infrastructure vuSpot (see Section 3.1) were used to build this game (see Figure A.1).

The client is responsible for receiving from the server the augmented video stream and displaying it. It is also responsible for sending to the server the user input and any information from sensors present in the device.

Besides communicating with the clients, the server is responsible for the communication with the video cameras of the video camera network and the database, that stores all the information needed for the system and contains several modules as well as all the video processing to determine the user's location and gestures performed.

The Communications Manager exchanges with the clients all information needed by the system.

The Location Module is used to detect other people in the location and their actions, objects of the space and virtual objects and characters.

The Video Stream Composer composes a video (using the relevant video stream of the location) with additional and relevant information about the space and other users. This information is received from the Location Module and Application Logic Manager.

The Gesture Detection Module detects user movements. This is necessary for the superimposing of reactions to those movements.

The Overlay sub-module overlays on the video the information relevant for the user visit,

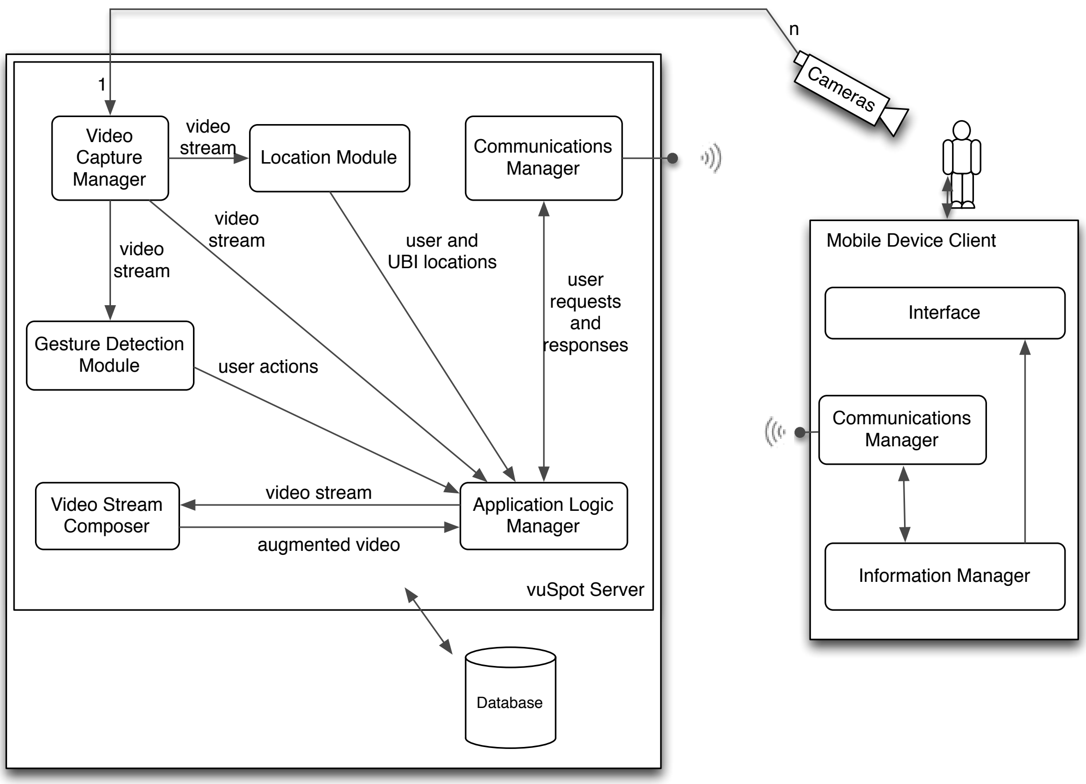


Figure A.1: UBI, The Guardian Dragon: The Infrastructure.

like information about sidekicks, points of interest in the space, virtual characters and virtual objects. The Application Logic Manager generates information about the context, user actions and interaction results. The resulting video is sent to the Communications Manager to be delivered to the client.

The client receives the video stream so that the user can visualise her sidekick and other virtual elements present near her location. Interaction can be done by using the input capabilities of the mobile device or by performing actions near the sidekick. This will trigger a reaction to be superimposed on the video stream.

Delivering video streams to mobile devices depends on the speed of the wireless networks and computational power of the mobile device. The video stream adapts to the mobile device capabilities dropping frames if needed. This strategy will lower the performance of the application but will make it possible to use with slower wi-fi networks and mobile devices.

A.2 MagicLight

In MagicLight (see Figure A.2), video streams are processed in order to extract information that is provided to the dæmon giving means for the dæmon to react to the user's actions. We used the video capture, detection and location modules. The system also includes a

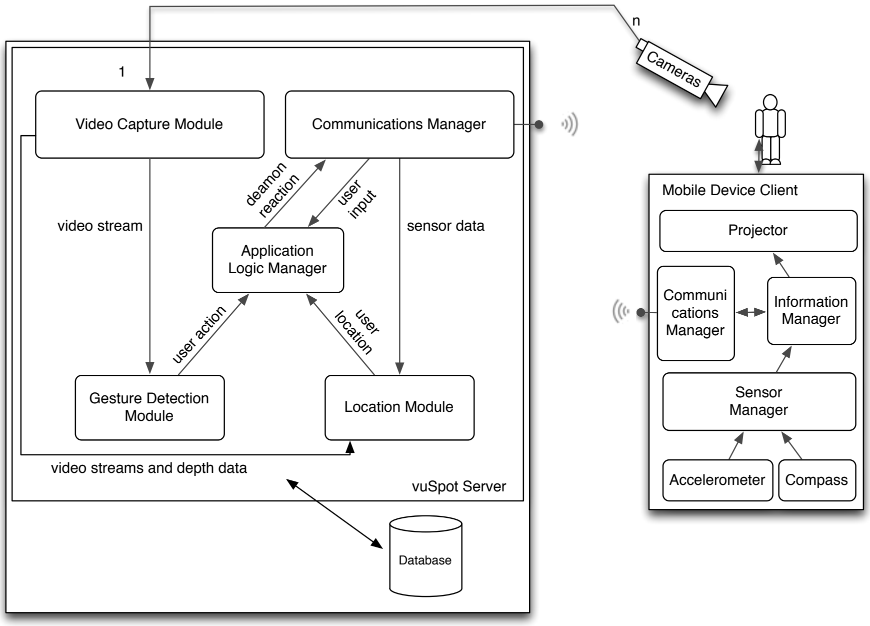


Figure A.2: MagicLight: The Infrastructure.

communication manager, an application logic manager and a database. All information received from the mobile device client, from the video camera network and that is generated by the system is stored in the database.
Interaction can be done directly with the MagicLight by actuating the accelerometer or compass or with gestures that are captured on video by the video camera network.
The information obtained by the sensor manager from the mobile device sensors is organised in the form of a XML document by the information manager. All information exchanged between the client and the server is sent or received by the respective communications manager. The application logic manager processes it or, in case of GPS data, the processing is done by the location module.
The video streams captured by the video camera network that exists at the space are sent to the video capture module. The captured video streams of the space are processed and the user's actions recognised using the Gesture Detection Module. Upper body gestures like waving can be recognised. Tracking of the user is made using the Location Module by inferring the user position from the video streams.
According to the user's action recognised, the user's input and the user's location, the reaction information is sent to the mobile device client application to be visualised. The XML document contains information about the user id, which dæmon image is the user's dæmon, which dæmon reaction is to be displayed, what is the location of the dæmon's reaction image in the display (C=center, L=left, R=right), the relaxation, shyness and adventure levels of the user inferred from the user profile. Some of this information could be passed to the mobile client only once at the beginning of the interaction but for evaluation purposes it is passed every time. This allows to change the user profile during

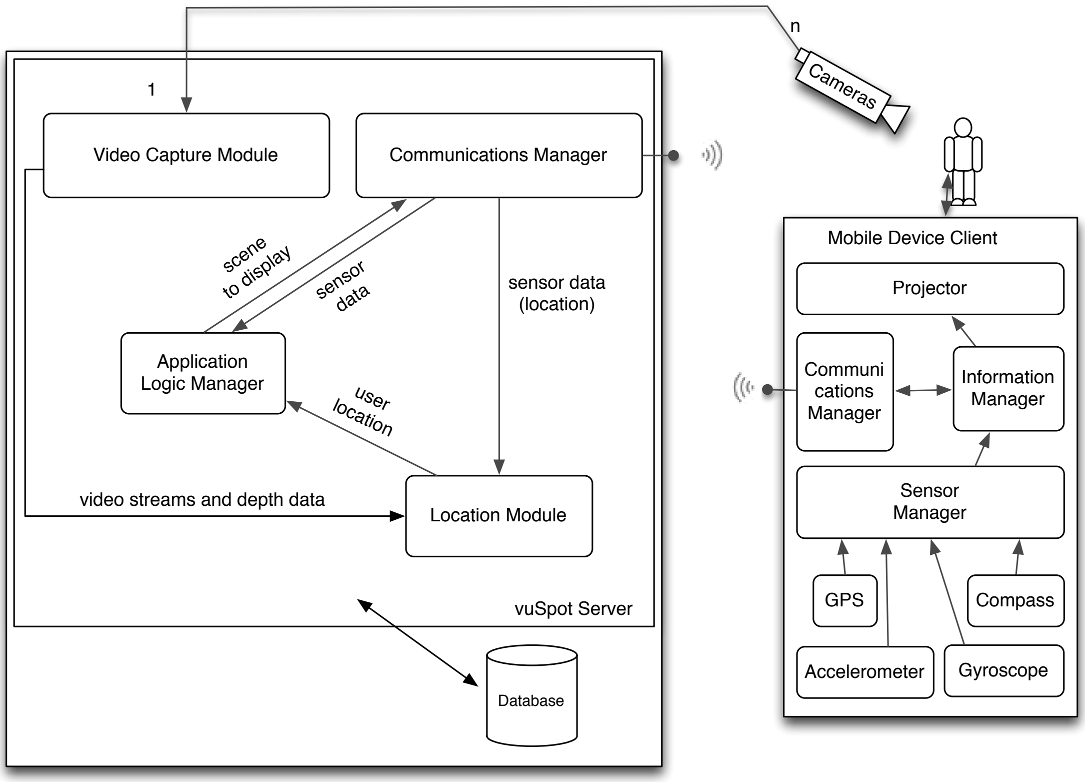


Figure A.3: Gone Fishing and Haunted House: The Infrastructure.

the interaction and check if the dæmon reacts accordingly. By sending XML to the mobile client instead of the actual image of the reaction of the dæmon, the performance of the application is increased.

The mobile device client receives the XML document and the information manager parses it obtaining the information needed to generate the dæmon response to be projected.

A.3 Gone Fishing and Haunted House

In GoneFishing and Haunted House, the video streams from the existing video camera networks (obtained and managed by the Video Capture Manager) are used to obtain information about the user's location by the Location Module. The Location Module is also responsible for locating the virtual elements present at the space.
The user performs actions using the mobile device. These actions actuate the mobile device sensors. The data from the sensors is gathered by the client's application Sensors Manager and processed by the client's application Information Manager. The relevant information is then sent to the server's Communication Manager using the client's Communication Manager. The information to be displayed is generated by the server's Application Logic Manager according to the information obtained by the server's Location Module and data

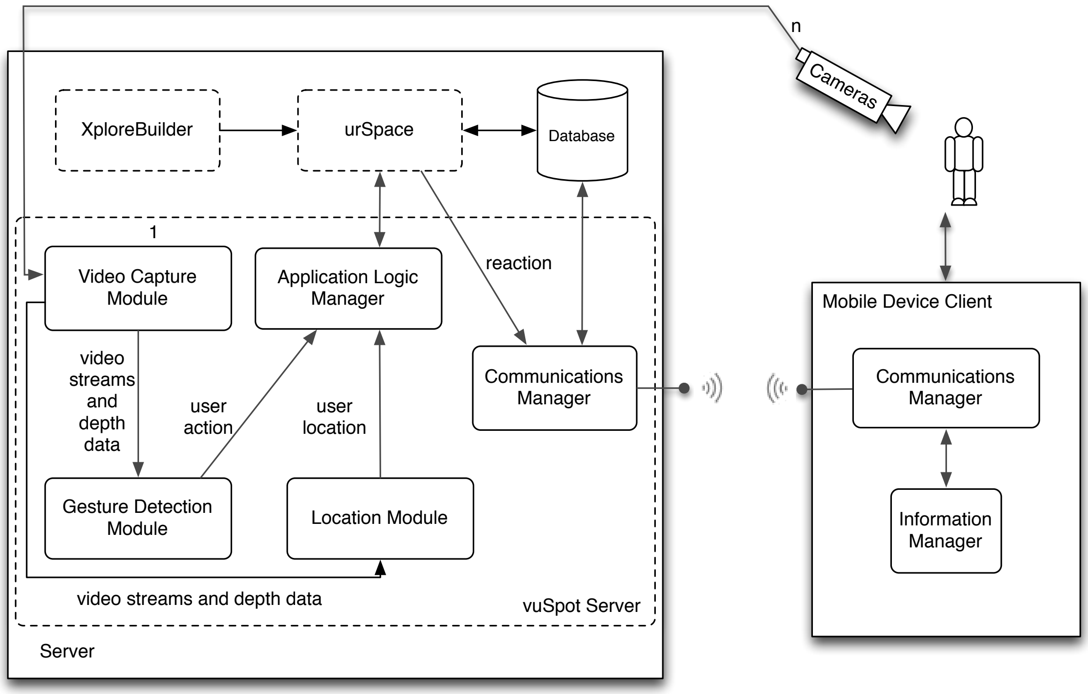


Figure A.4: Toy Exhibition: The Infrastructure.

from mobile device sensors. This generated information is sent to the mobile device (using the Communications Managers) and displayed by the mobile device's projector.

A.4 Toy Exhibition Application Example

Several modules of the infrastructure vuSpot (see Section 3.1) were used to build this application (see Figure A.4). For the purpose of the application we used the video capture, detection and location modules. The system also includes a communication manager and a database.

The space description file as well as the interactive experience application description file (which were generated by the XploreBuilder visual tool) are used to configure the system. From the interactive experience application description file, the information about interactions can be obtain, that is, what reaction is to be visualised by the visitor when a given action is performed in the surroundings of a given Point Of Interest.

The video streams captured by the video camera network that exists at the space are sent to the Video Capture Module.

Video streams are processed in order to extract information about the actions the user performs in the neighbourhood of a Point Of Interest.

The captured video streams of the space are processed by the Gesture Detection Module and the user's actions recognised.

User actions must be performed in the neighbourhood of a Point Of Interest, which means that the location of the user must be accurate.
The Location Module is responsible for tracking the user. The user location is obtained by inferring the user position from the video streams.
According to the user's location and action recognised, the corresponding reaction is generated. A reaction might be text or an image to display, a sound to play or to activate the vibrate mode of the mobile device (provided that the mobile device has this capability).
The information generated is sent by the Communications Manager to the mobile device client application with the indication of the reaction expected.
The responsibilities of the mobile client application are:

1. Receive messages from the server with the reactions expected.
2. Execute the reaction described in the received messages.
3. End the interaction when a finish message is received.

APPENDIX B

XPLOREDESCRIPTION GRAMMARS DESCRIPTION AND USAGE

The description of the lexical and language grammars of Listing 3.3 are presented in Section B.1 and B.2 respectively. In the last section we describe the usage of the lexical and language grammars.

B.1 Description of the lexical grammar

We present a description of the DSL's lexical grammar of Listing 3.3 in Table B.1.

Lines	Grammar	Description
1	%lex	Beginning of the lexical grammar
2		
3	%%	
4	\s+	Ignore whitespaces
5		
6	[0-9]+("."[0-9]+)? \b	NUMBER definition
7		
8	/* UNIVERSAL POIs */	Points Of Interest that exist in every space
9	"NO_POI"	Regarding Points-Of-Interest: NO POI in the surroundings of the user or pointing the device at NO POI
10	"START"	Regarding Points-Of-Interest: START of the visit is considered a Point-Of-Interest
11		
12	/* TIME */	Time definitions

13	"WAIT"	Timings regarding visualisation: WAIT a NUMBER of seconds before displaying
14	"DURATION"	Timings regarding visualisation: display for a DURATION of a NUMBER of seconds
15	"NONE"	Timings regarding visualisation: NONE defined
16	"FOREVER"	Timings regarding visualisation: visualise FOREVER or until another FOREVER item needs to be visualised
17	"MOUSE_CLICK"	Timings regarding visualisation: visualise until MOUSE CLICK
18		
19	/* BODY GESTURES */	Actions performed by the user using the body to perform gestures
20	"WAVE"	User body gesture related actions: user WAVEs
21	"RAISE_HAND"	User body gesture related actions: user RAISEs HAND
22	"POKE"	User body gesture related actions: user performs the POKE gesture
23	"CLICK"	User body gesture related actions: user performs the CLICK gesture
24	"TAP"	User body gesture related actions: user performs the TAP gesture
25		
26	/* DEVICE GESTURES */	Actions performed by the user using the mobile device to perform gestures
27	"SHAKE"	Device gesture related actions: user SHAKEs the device
28	"TURN_DOWN"	Device gesture related actions: user TURNs DOWN to the floor the device
29	"TURN_UP"	Device gesture related actions: user TURNs UP to the ceiling the device
30	"POINT"	Device gesture related actions: user POINTs the device in some direction
31		
32	/* MOVEMENT ACTIONS */	Actions performed by the movement and location of the user
33	"STOP"	Movements related actions: user is STOPped
34	"OUT_OF_VIEW"	Movements related actions: user is moving OUT OF VIEW

35	"MOVING_AWAY"	Movements related actions: user is MOVING AWAY from a location
36	"MOVING_TO"	Movements related actions: user is MOVING TO in direction of a location
37		
38	/* GENERIC REACTIONS */	Reactions found in every application
39	"CHANGE_IMAGE"	Reactions: CHANGE the IMAGE being visualised even if it has a FOREVER timing indication
40	"VIBRATE"	Reactions: VIBRATE the mobile device
41		
42	/* COLLECTING */	Collections
43	"COLLECT"	Regarding collections: item is to be COLLECTed
44	"DO_NOT_COLLECT"	Regarding collections: DO NOT COLLECT the item
45	"SHOW_COLLECTION"	Regarding collections: SHOW COLLECTION of collected items
46		
47	"->"	Operators
48	"OK->"	
49	"NOK->"	
50	"WITH"	
51	"="	
52	":"	
53	"{"	
54	"}"	
55	","	
56	"AND"	
57	"OR"	
58	"WITH"	
59	"IFPROJECTOR"	
60	"IFNOPROJECTOR"	
61		
62	/lex	End of the lexical grammar definition

Table B.1: Description of the lexical grammar of Listing 3.3

B.2 Description of the language grammar

In Table B.2, we present a description of the DSL's language grammar of Listing 3.3.

Lines	Grammar	Description
63		
64	* operator associations and precedence *	Operator associations and precedence
65	%nonassoc '{' '}'	
66	%right '='	
67	%left ':'	
68	%left '->'	
69	%left 'AND' 'OR'	
70	%right ','	
71	%left '|'	
72		
73	%start SPACE_EXPLORATION	Starting rule: SPACE_EXPLORATION
74		
75	* language grammar *	
76	SPACE_EXPLORATION : GAME	Rule: SPACE EXPLORATION can be in the form of a GAME, an INTERACTIVE TOUR or a simple TOUR (just a sequence of POI with no interactions defined)
77	| INTERACTIVE_TOUR	
78	| TOUR;	
79		
80	GAME : '{' INTERACTION_SEQUENCE '}' ',' FINISH	Rule: A GAME is described by a set of INTERACTION SEQUENCES and has as a FINISH
81	| IFPROJECTOR '{' '{' INTERACTION_SEQUENCE '}' ',' FINISH '}'	Rule: A GAME can have two different descriptions. One IF a PROJECTOR is present and another IF NO PROJECTOR is present
82	IFNOPROJECTOR '{' '{' INTERACTION_SEQUENCE '}' ',' FINISH '}';	
83		
84	INTERACTIVE_TOUR : '{' INTERACTION_SEQUENCE '}'	Rule: An INTERACTIVE TOUR is described by a set of INTERACTION SEQUENCES and has no finish
85	| IFPROJECTOR '{' '{' INTERACTION_SEQUENCE '}' '}'	Rule: An INTERACTIVE TOUR can have two different descriptions. One IF a PROJECTOR is present and another IF NO PROJECTOR is present
86	IFNOPROJECTOR '{' '{' INTERACTION_SEQUENCE '}' '}';	

87		
88	TOUR : '{' POI_SEQUENCE '}'	Rule: A TOUR is described by a set of Points Of Interest and has no finish
89	| IFPROJECTOR '{' '{' POI_SEQUENCE '}' '}'	Rule: A TOUR can have two different descriptions. One IF a PROJECTOR is present and another IF NO PROJECTOR is present
90	IFNOPROJECTOR '{' '{' POI_SEQUENCE '}' '}';	
91		
92	FINISH : INTERACTION_SEQUENCE ;	Rule: the FINISH of a GAME is described as an INTERACTION SEQUENCE
93		
94	VIRTUAL_LOCATION : VIRTUAL_CHARACTER | VIRTUAL_OBJECT ;	Rule: A VIRTUAL LOCATION can be a VIRTUAL CHARACTER or a VIRTUAL OBJECT
95		
96	POI_SEQUENCE : POI	Rule: A Point Of Interest SEQUENCE can be a Point Of Interest or be composed by more than one POI SEQUENCE
97	| POI_SEQUENCE ',' POI_SEQUENCE ;	
98		
99	INTERACTION_SEQUENCE : POI ':' ACTION TIME '->' REACTION TIME	Rule: An INTERACTION SEQUENCE can be composed of a Point Of Interest where an ACTION is performed with a given TIME condition. When the ACTION is performed a REACTION is visualised with a TIME condition
100	| POI ':' ACTION GAMEPLAY TIME '->' REACTION TIME	Rule: An INTERACTION SEQUENCE can be composed of a Point Of Interest where an ACTION is performed with a given GAMEPLAY condition and a TIME condition. When the ACTION is performed a REACTION is visualised with a TIME condition
101	| POI ':' '{' ACTION_SET '}' TIME '->' REACTION TIME	Rule: An INTERACTION SEQUENCE can be composed of a Point Of Interest where a SET of ACTIONs are performed with a given TIME condition. When the SET of ACTIONs are performed a REACTION is visualised with a TIME condition

102	| VIRTUAL_LOCATION ':' ACTION TIME '->' REACTION TIME	Rule: An INTERACTION SEQUENCE can be composed of a VIRTUAL LOCATION where an ACTION is performed with a given TIME condition. When the ACTION is performed a REACTION is visualised with a TIME condition
103	| VIRTUAL_LOCATION ':' '{' ACTION_SET '}' TIME '->' REACTION TIME	Rule: An INTERACTION SEQUENCE can be composed of a VIRTUAL LOCATION where a SET of ACTIONs are performed with a given TIME condition. When the SET of ACTIONs are performed a REACTION is visualised with a TIME condition
104	| '{' INTERACTION_SEQUENCE ',' INTERACTION_SEQUENCE '}'	Rule: An INTERACTION SEQUENCE can be composed of more than one INTERACTION SEQUENCE that have to be executed in the given order
105	| '{' INTERACTION_SET ',' INTERACTION_SEQUENCE '}'	Rule: An INTERACTION SEQUENCE can be composed of an INTERACTION SET and an INTERACTION SEQUENCE
106	| INTERACTION_SEQUENCE ',' INTERACTION_SEQUENCE;	Rule: An INTERACTION SEQUENCE can be composed of more than one INTERACTION SEQUENCE
107		
108	INTERACTION_SET : INTERACTION_SEQUENCE 'OR' INTERACTION_SEQUENCE	Rule: An INTERACTION SET can be composed of two INTERACTION SEQUENCES, where only one needs to be executed
109	| INTERACTION_SEQUENCE 'AND' INTERACTION_SEQUENCE;	Rule: An INTERACTION SET can be composed of two mandatory INTERACTION SEQUENCES that have to be both executed
110		
111	TIME :	
112	| NONE	Rule: TIME can be empty or NONE if not defined
113	| MOUSE_CLICK	Rule: TIME can be MOUSE CLICK for visualisation to occur until a mouse click or a tap on a touch mobile device occurs

114	| FOREVER	Rule: TIME can be FOREVER when visualisation will last throughout the execution time of the application or until another FOREVER item needs to be visualised
115	| DURATION ′=′ NUMBER	Rule: TIME can have a DURATION of the NUMBER of seconds the item is to remain in visualisation
116	| WAIT ′=′ NUMBER	Rule: TIME can have a WAIT of the NUMBER of seconds before the item is visualised
117	| DURATION ′=′ NUMBER WAIT ′=′ NUMBER;	Rule: TIME can have a DURATION of the NUMBER of seconds the item is to remain in visualisation and a WAIT of the NUMBER of seconds before the item is visualised
118		
119	ACTION : BODY_ACTION	Rule: an ACTION can be a BODY ACTION, a MOVEMENT ACTION or a DEVICE ACTION
120	| MOVEMENT_ACTION	
121	| DEVICE_ACTION;	
122		
123	ACTION_SET : MOVEMENT_ACTION ′,′ BODY_ACTION	Rule: an ACTION SET can be composed by a combination of actions: a MOVEMENT ACTION and a BODY ACTION
124	| MOVEMENT_ACTION ′,′ DEVICE_ACTION	Rule: an ACTION SET can be composed by a combination of actions: a MOVEMENT ACTION and a DEVICE ACTION
125	| POINT ′,′ DEVICE_ACTION	Rule: an ACTION SET can be composed by a combination of actions: a POINTing action and a DEVICE ACTION
126	| BODY_ACTION ′,′ DEVICE_ACTION;	Rule: an ACTION SET can be composed by a combination of actions: a BODY ACTION and a DEVICE ACTION
127		
128	REACTION : SHOW_REACTION	Rule: a REACTION can be to SHOW a REACTION
129	| SHOW_REACTION COLLECTION	Rule: a REACTION can be to SHOW a REACTION indicating if it is for a COLLECTION
130	| QUESTIONING	Rule: a REACTION can be QUESTIONING
131	| INPUT	Rule: a REACTION can be a request for INPUT

132	| SHOW_COLLECTION	Rule: a REACTION can be to SHOW a COLLECTION
133	| SHOW_REACTION CHANGE_IMAGE;	Rule: a REACTION can be to SHOW a REACTION and also CHANGE the IMAGE being displayed
134	| VIDEO ′ WITH ′ SHOW_REACTION	Rule: a VIDEO stream can be AUGMENTED WITH a REACTION
135		
136	QUESTIONING : QUESTION ANSWER	Rule: QUESTIONING is composed of a QUESTION and an ANSWER
137	OK->′ SHOW_REACTION COLLECTION	Rule: if user answer is OK, meaning it is the ANSWER, then SHOW REACTION indicating if it is for COLLECTION
138	NOK->′ SHOW_REACTION COLLECTION;	Rule: if user answer is Not OK, meaning it is not the ANSWER, then SHOW REACTION indicating if it is for COLLECTION
139		
140	COLLECTING : | COLLECT | NOT_COLLECT;	Rule: COLLECTING indication can be undefined, to COLLECT or to NOT COLLECT.

Table B.2: Description of the language grammar of Listing 3.3

B.3 Grammar usage

In the next table we describe when the several components of the grammar can be used.

Lines	Grammar	Usage
1	%lex	
2		
3	%%	
4	\s+	
5		
6	[0-9]+("."[0-9]+)? \b	When time in seconds needs to be defined.
7		
8	/* UNIVERSAL POIs */	
9	"NO_POI"	When the user performs actions and there is not a POI in the neighbourhood or the user's mobile device is not pointing at one, there might be the need to define a reaction.
10	"START"	START is the location where the user started the application and it is treated as a POI. When the application starts some information can be displayed (such as a title) regardless of POIs in the neighbourhood of the user.
11		
12	/* TIME */	
13	"WAIT"	When there is the need to delay the visualisation for a given number os seconds
14	"DURATION"	When there is the need to define the duration of the visualisation of an item.
15	"NONE"	When we do not want to define timings for the visualisation.
16	"FOREVER"	When there is the need to visualise something for the whole duration of the application execution (such as a background image). If another item marked with FOREVER is displayed the first one is substituted.
17	"MOUSE_CLICK"	When there is the need to visualise an item until there is a mouse click or a tap (in case of a touch mobile device).
18		
19	/* BODY GESTURES */	

20	"WAVE"	Recognisable gestures that a a user can perform with her body. These gestures are used as actions to trigger reactions.
21	"RAISE_HAND"	
22	"POKE"	
23	"CLICK"	
24	"TAP"	
25		
26	/* DEVICE GESTURES */	
27	"SHAKE"	Recognisable gestures that a user can perform with her mobile device (detected using the mobile device sensors). These gestures are used as actions to trigger reactions.
28	"TURN_DOWN"	
29	"TURN_UP"	
30	"POINT"	
31		
32	/* MOVEMENT ACTIONS */	When the reactions depend on the location of the user.
33	"STOP"	STOP means the user is stopped.
34	"OUT_OF_VIEW"	OUT_OF_VIEW means the user is out of view from the video cameras of the video camera network.
35	"MOVING_AWAY"	MOVING_AWAY is associated with a POI and is used when there is the need to trigger a reaction because the user is moving away from that POI.
36	"MOVING_TO"	MOVING_TO is associated with a POI and is used when there is the need to trigger a reaction because the user is moving in the direction of that POI.
37		
38	/* GENERIC REACTIONS */	When there is an action, a reaction is triggered. The reaction depends on the application. The following two reactions are common with all applications.
39	"CHANGE_IMAGE"	CHANGE_IMAGE changes the image being visualised. CHANGE_IMAGE is used when there is the need to change the image related to an item to a new one.
40	"VIBRATE"	VIBRATE makes the mobile device vibrate.
41		

42	/* COLLECTING */	When the application allows the collection of items.
43	"COLLECT"	An item can have the indication of COLLECT (can be collected)
44	"DO_NOT_COLLECT"	An item can have the indication of DO_NOT_COLLECT (can not be collected).
45	"SHOW_COLLECTION"	The items collected can be shown using SHOW_COLLECTION.
46		
47	"->"	
48	"OK->"	
49	"NOK->"	
50	"WITH"	
51	"="	
52	":"	
53	"{"	
54	"}"	
55	","	
56	"AND"	
57	"OR"	
58	"WITH"	
59	"IFPROJECTOR"	
60	"IFNOPROJECTOR"	
61		
62	/lex	
63		
64	* operator associations and precedence *	
65	%nonassoc '{' '}'	
66	%right '='	
67	%left ':'	
68	%left '->'	
69	%left 'AND' 'OR'	
70	%right ','	
71	%left '|'	
72		
73	%start SPACE_EXPLORATION	
74		
75	* language grammar *	

76	SPACE_EXPLORATION : GAME	A Space exploration application can be of three types: GAME (which is a set of sequences describing interaction and has one or more finishing sequences), an INTERACTIVE_TOUR (which is a game where the finish is not defined) and a TOUR (which is a set of Points Of Interest to be visited). The three types allow to define the behaviour of the application if the mobile device has projection capabilities or not. In case of not defining the visualisation is the same both in the display and projection.
77	| INTERACTIVE_TOUR	
78	| TOUR;	
79		
80	GAME : ’{’ INTERACTION_SEQUENCE ’}’ ’,’ FINISH	
81	| IFPROJECTOR ’{’ ’{’ INTERACTION_SEQUENCE ’}’ ’,’ FINISH ’}’	
82	IFNOPROJECTOR ’{’ ’{’ INTERACTION_SEQUENCE ’}’ ’,’ FINISH ’}’;	
83		
84	INTERACTIVE_TOUR : ’{’ INTERACTION_SEQUENCE ’}’	
85	| IFPROJECTOR ’{’ ’{’ INTERACTION_SEQUENCE ’}’ ’}’	
86	IFNOPROJECTOR ’{’ ’{’ INTERACTION_SEQUENCE ’}’ ’}’;	
87		
88	TOUR : ’{’ POI_SEQUENCE ’}’	
89	| IFPROJECTOR ’{’ ’{’ POI_SEQUENCE ’}’ ’}’	
90	IFNOPROJECTOR ’{’ ’{’ POI_SEQUENCE ’}’ ’}’;	
91		
92	FINISH : INTERACTION_SEQUENCE ;	A game has to have at least one interaction sequence representing the finish.
93		
94	VIRTUAL_LOCATION : VIRTUAL_CHARACTER | VIRTUAL_OBJECT ;	When there is the need to perform action in the neighbourhood of a virtual character or object a virtual location is used.
95		
96	POI_SEQUENCE : POI	A POI SEQUENCE is a set of at least two POIs.
97	| POI_SEQUENCE ’,’ POI_SEQUENCE ;	
98		

99	INTERACTION_SEQUENCE : POI ':' ACTION TIME '->' REACTION TIME	INTERACTION SEQUENCE simplest form using a POI.
100	| POI ':' ACTION GAMEPLAY TIME '->' REACTION TIME	An action can have a gameplay condition.
101	| POI ':' '{' ACTION_SET '}' TIME '->' REACTION TIME	A reaction can be triggered by a SET of ACTIONS.
102	| VIRTUAL_LOCATION ':' ACTION TIME '->' REACTION TIME	INTERACTION SEQUENCE simplest form using a VIRTUAL LOCATION.
103	| VIRTUAL_LOCATION ':' '{' ACTION_SET '}' TIME '->' REACTION TIME	A reaction can be triggered by a SET of ACTIONS.
104	| '{' INTERACTION_SEQUENCE ',' INTERACTION_SEQUENCE '}'	An INTERACTION_SEQUENCE can be defined by a set of INTERACTION_SEQUENCE that need to be executed in order from left to right.
105	| '{' INTERACTION_SET ',' INTERACTION_SEQUENCE '}'	An INTERACTION_SEQUENCE can be defined by an INTERAC- TION_SET followed by an INTERACTION_SEQUENCE.
106	| INTERACTION_SEQUENCE ',' INTERACTION_SEQUENCE;	An INTERACTION_SEQUENCE can be defined by a set of INTERACTION_SEQUENCE.
107		
108	INTERACTION_SET : INTERACTION_SEQUENCE 'OR' INTERACTION_SEQUENCE	An INTERACTION_SET is used when two or more INTERACTION_SEQUENCE need to all be executed (AND) or when at least one of the set needs to be executed (OR) before the INTERACTION_SEQUENCE.
109	| INTERACTION_SEQUENCE 'AND' INTERACTION_SEQUENCE;	
110		
111	TIME :	When there is the need to define the timing the visualisation of an item (delay, duration or disappear). May be empty.
112	| NONE	
113	| MOUSE_CLICK	
114	| FOREVER	
115	| DURATION '=' NUMBER	
116	| WAIT '=' NUMBER	
117	| DURATION '=' NUMBER WAIT '=' NUMBER;	

118		
119	ACTION : BODY_ACTION	Used to define the type of actions and combinations of actions. POINT is used when the user is pointing the mobile device at some direction.
120	| MOVEMENT_ACTION	
121	| DEVICE_ACTION;	
122		
123	ACTION_SET : MOVEMENT_ACTION ',' BODY_ACTION	
124	| MOVEMENT_ACTION ',' DEVICE_ACTION	
125	| POINT ',' DEVICE_ACTION	
126	| BODY_ACTION ',' DEVICE_ACTION;	
127		
128	REACTION : SHOW_REACTION	Used to define the types of reaction.
129	| SHOW_REACTION COLLECTION	
130	| QUESTIONING	
131	| INPUT	
132	| SHOW_COLLECTION	
133	| SHOW_REACTION CHANGE_IMAGE	
134	| VIDEO ' WITH ' SHOW_REACTION;	
135		
136	QUESTIONING : QUESTION ANSWER	When there is the need to ask questions to the user. The QUESTION is define as well as the respective ANSWER. Reactions for the correct answer and incorrect answer can de defined.
137	OK->' SHOW_REACTION COLLECTION	
138	NOK->' SHOW_REACTION COLLECTION;	
139		
140	COLLECTING : | COLLECT | NOT_COLLECT;	When there is the need to indicate the ability of collecting items (POIs).

Table B.3: Usage examples for the grammar in Listing 3.3

APPENDIX

C

Application Description Examples

Five case studies where the DSL's grammar was used are presented in the next sections. These case studies correspond to the applications developed during this research and that are described in Section 4. For each case study the space description file, the grammar specific information for each space and the application description file are presented.

C.1 UBI, The Guardian Dragon

Listing C.1: UBI, The Guardian Dragon application space description.

```
Space Name
Famous People
Description
UBI, The Guardian Dragon, your virtual sidekick, helps explore the Famous People Exhibition
Number of Rooms
1
Room1
Name
Famous People
Description
Famous Five
Width
5
Height
4
Length
15
Cameras
CAMERA1,0,Webcam,0.0,0.0,1.0,43.0,57.0
CAMERA2,1,Kinect,2.5,0.0,1.0,65.0,80.0
POIs
POI11,Steve Jobs,STEVE_JOBS,,2,15,1.5
POI12,Dom Quixote,DOM_QUIXOTE,,2,0,1.5
POI13,Pythagoras,PYTHAGORAS,,0,13,1.5
POI14,Joan Miro,JOAN_MIRO,,4,0,1.5
POI15,Luis Vaz de Camoes,LUIS_VAZ_DE_CAMOES,,3,10,1.5
Virtual Elements
POI16,UBI,UBI,,Character,0.0,0.0,0.0,100,170,center,0.0,0.0,0.0
Actions
```

```
Wave
Poke
Moving Away
Moving To
Stop
Texts
,,,,,,,,
Multimedia
IMAGE101,Image,How many triangles?,How many triangles?,POI103,SPACE2/pythagoras_hint_ballon.png,false,
     false,,
IMAGE102,Image,Hint: look at the map,Hint: look at the map,,SPACE2/finish_hint_ballon.png,false,false
     ,,
IMAGE103,Image,Breath Fire,Breath Fire,,SPACE2/dragon_fire.png,false,false,,
IMAGE104,Image,Wage Tail Down,Wage Tail Down,,SPACE2/dragon_headT_tailD.png,false,false,,
IMAGE105,Image,Wage Tail Up,Wage Tail Up,,SPACE2/dragon_headT_tailU.png,false,false,,
IMAGE106,Image,Flap Wing Down - Tail Down,Flap Wing Down - Tail Down,,SPACE2/dragon_tailD_wingD.png,
     false,false,,
IMAGE107,Image,Flap Wing Up - Tail Down,Flap Wing Up - Tail Down,,SPACE2/dragon_tailD.png,false,false
     ,,
IMAGE108,Image,Guess the word: 8 letters,Guess the word: 8 letters,,SPACE2/start_ballon.png,false,
     false,,
IMAGE109,Image,Hint: iPod, iPad, iPhone, iMac,Hint: iPod, iPad, iPhone, iMac,POI101,SPACE2/
     jobs_hint_ballon.png,false,false,,
IMAGE110,Image,Correct! You have earned the letter: i,Correct! You have earned the letter: i,POI101,
     SPACE2/letter_I_ballon.png,true,false,TEXT,I
IMAGE111,Image,Second letter of Portugal's neighbour.,Second letter of Portugal's neighbour.,POI104,
     SPACE2/miro_hint_ballon.png,false,false,,
IMAGE112,Image,Correct! You have earned the letter: p,Correct! You have earned the letter: p,POI104,
     SPACE2/letter_P_ballon.png,true,false,TEXT,P
IMAGE113,Image,Correct! You have earned the letter: l,Correct! You have earned the letter: l,POI103,
     SPACE2/letter_L_ballon.png,true,false,TEXT,L
IMAGE114,Image,Hint: Look for the name of the farmer on the third paragraph.,Hint: Look for the name
     of the farmer on the third paragraph.,POI102,SPACE2/quixote_hint_ballon.png,false,false,,
IMAGE115,Image,Correct! You have earned the letter: o,Correct! You have earned the letter: o,POI102,
     SPACE2/letter_O_ballon.png,true,false,TEXT,O
IMAGE116,Image,Wrong! Try again!,Wrong! Try again!,,SPACE2/wrong_ballon.png,false,false,,
IMAGE117,Image,THE END! Hope you enjoyed the game. Ask for your prize!,THE END! Hope you enjoyed the
     game. Ask for your prize!,POI105,SPACE2/camoes_ballon.png,false,false,,
IMAGE118,Image,You have all the letters. Find Camoes!,You have all the letters. Find Camoes!,,SPACE2/
     finish_ballon.png,false,false,,
IMAGE119,Image,Map,Map,,SPACE2/map.png,false,false,,
IMAGE120,Image,Background Image,Background Image,Start,SPACE2/background.png,false,false,,
VIDEO101,Video,Video,Video,,,false,false,,
Questions
QUESTION101,What is apple's most famous letter?,,c,n,u,i,POI101
QUESTION102,What's the second letter of the author's nationality in english?,,a,p,a,s,POI104
QUESTION103,How many right-angled triangles are there in Figure 2?,triangles.jpg,a,44,48,40,POI103
QUESTION104,What is the last letter of the first name of Quixote's companion?,,b,s,o,z,POI102
Requests
,,,,,,,
```

Listing C.2: UBI, The Guardian Dragon application grammar.

```
%lex

%%
\s+ {/* skip whitespace */}

[0-9]+("."[0-9]+)?\b {return 'NUMBER';}


/* UNIVERSAL POINTS OF INTEREST */
"NO_POI" {return 'POI';}
"START" {return 'POI';}

/* UBI - POINTS OF INTEREST */
"POI101" {return 'POI';}
"POI102" {return 'POI';}
```

```
"POI103" {return 'POI';}
"POI104" {return 'POI';}
"POI105" {return 'POI';}
"POI106" {return 'VIRTUAL_CHARACTER';}

/* TIME */
"WAIT" {return 'WAIT';}
"DURATION" {return 'DURATION';}
"NONE" {return 'NONE';}
"FOREVER" {return 'FOREVER';}
"MOUSE_CLICK" {return 'MOUSE_CLICK';}

/* BODY GESTURES */
"WAVE" {return 'BODY_ACTION';}
"RAISE_HAND" {return 'BODY_ACTION';}
"POKE" {return 'BODY_ACTION';}
"CLICK" {return 'BODY_ACTION';}
"TAP" {return 'BODY_ACTION';}

/* DEVICE GESTURES */
"SHAKE" {return 'DEVICE_ACTION';}
"TURN_DOWN" {return 'DEVICE_ACTION';}
"TURN_UP" {return 'DEVICE_ACTION';}
"POINT" {return 'POINT';}

/* MOVEMENT ACTIONS */
"STOP" {return 'MOVEMENT_ACTION';}
"OUT_OF_VIEW" {return 'MOVEMENT_ACTION';}
"MOVING_AWAY" {return 'MOVEMENT_ACTION';}
"MOVING_TO" {return 'MOVEMENT_ACTION';}

/* GENERIC REACTIONS */
"CHANGE_IMAGE" {return 'CHANGE_IMAGE';}
"VIBRATE" {return 'SHOW_REACTION';}

/* UBI - REACTIONS */
"IMAGE101" {return 'SHOW_REACTION';}
"IMAGE102" {return 'SHOW_REACTION';}
"IMAGE103" {return 'SHOW_REACTION';}
"IMAGE104" {return 'SHOW_REACTION';}
"IMAGE105" {return 'SHOW_REACTION';}
"IMAGE106" {return 'SHOW_REACTION';}
"IMAGE107" {return 'SHOW_REACTION';}
"IMAGE108" {return 'SHOW_REACTION';}
"IMAGE109" {return 'SHOW_REACTION';}
"IMAGE110" {return 'SHOW_REACTION';}
"IMAGE111" {return 'SHOW_REACTION';}
"IMAGE112" {return 'SHOW_REACTION';}
"IMAGE113" {return 'SHOW_REACTION';}
"IMAGE114" {return 'SHOW_REACTION';}
"IMAGE115" {return 'SHOW_REACTION';}
"IMAGE116" {return 'SHOW_REACTION';}
"IMAGE117" {return 'SHOW_REACTION';}
"IMAGE118" {return 'SHOW_REACTION';}
"IMAGE119" {return 'SHOW_REACTION';}
"IMAGE120" {return 'SHOW_REACTION';}
"VIDEO101" {return 'VIDEO';}

/* Q and A */
"QUESTION101" {return 'QUESTION';}
"QUESTION102" {return 'QUESTION';}
"QUESTION103" {return 'QUESTION';}
"QUESTION104" {return 'QUESTION';}

"ANSWER101" {return 'ANSWER';}
"ANSWER102" {return 'ANSWER';}
"ANSWER103" {return 'ANSWER';}
"ANSWER104" {return 'ANSWER';}
```

```
/* COLLECTING */
"COLLECT" {return 'COLLECT';}
"DO_NOT_COLLECT" {return 'NOT_COLLECT';}
"SHOW_COLLECTION" {return 'SHOW_COLLECTION';}

"->" {return '->';}
"OK->" {return 'OK->';}
"NOK->" {return 'NOK->';}
"WITH" {return 'WITH';}
"=" {return '=';}
":" {return ':';}
"{" {return '{';}
"}" {return '}';}
"," {return ',';}
"AND" {return 'AND';}
"OR" {return 'OR';}
"IFPROJECTOR" {return 'IFPROJECTOR';}
"IFNOPROJECTOR" {return 'IFNOPROJECTOR';}

/lex

/* operator associations and precedence */

%nonassoc '{' '}'
%right '='
%left ':'
%left '->'
%left 'AND' 'OR'
%right ','
%left '|'

%start SPACE_EXPLORATION

%% /* language grammar */

SPACE_EXPLORATION : GAME
                               | INTERACTIVE_TOUR
                               | TOUR;

GAME : '{' INTERACTION_SEQUENCE '}' ',' FINISH
         | IFPROJECTOR '{' '{' INTERACTION_SEQUENCE '}' ',' FINISH '}'
          IFNOPROJECTOR '{' '{' INTERACTION_SEQUENCE '}' ',' FINISH '}';

INTERACTIVE_TOUR : '{' INTERACTION_SEQUENCE '}'
                            | IFPROJECTOR '{' '{' INTERACTION_SEQUENCE '}' '}'
                             IFNOPROJECTOR '{' '{' INTERACTION_SEQUENCE '}' '}';

TOUR : '{' POI_SEQUENCE '}'
        | IFPROJECTOR '{' '{' POI_SEQUENCE '}' '}'
         IFNOPROJECTOR '{' '{' POI_SEQUENCE '}' '}';

FINISH : INTERACTION_SEQUENCE ;

VIRTUAL_LOCATION : VIRTUAL_CHARACTER | VIRTUAL_OBJECT ;

POI_SEQUENCE : POI
                       | POI_SEQUENCE ',' POI_SEQUENCE ;

INTERACTION_SEQUENCE : POI ':' ACTION TIME '->' REACTION TIME
                | POI ':' ACTION GAMEPLAY TIME '->' REACTION TIME
                | POI ':' '{' ACTION_SET '}' TIME '->' REACTION TIME
                | VIRTUAL_LOCATION ':' ACTION TIME '->' REACTION TIME
                | VIRTUAL_LOCATION ':' '{' ACTION_SET '}' TIME '->' REACTION TIME
                | '{' INTERACTION_SEQUENCE ',' INTERACTION_SEQUENCE '}'
                | '{' INTERACTION_SET ',' INTERACTION_SEQUENCE '}'
                | INTERACTION_SEQUENCE ',' INTERACTION_SEQUENCE;

INTERACTION_SET : INTERACTION_SEQUENCE 'OR' INTERACTION_SEQUENCE
```

```
                            | INTERACTION_SEQUENCE 'AND' INTERACTION_SEQUENCE;

TIME :
    | NONE
    | MOUSE_CLICK
    | FOREVER
    | DURATION '=' NUMBER
    | WAIT '=' NUMBER
    | DURATION '=' NUMBER WAIT '=' NUMBER;

ACTION : BODY_ACTION
           | MOVEMENT_ACTION
           | DEVICE_ACTION;

ACTION_SET : MOVEMENT_ACTION ',' BODY_ACTION
                  | MOVEMENT_ACTION ',' DEVICE_ACTION
                  | POINT ',' DEVICE_ACTION
                  | BODY_ACTION ',' DEVICE_ACTION;

REACTION : SHOW_REACTION
               | SHOW_REACTION COLLECTING
               | QUESTIONING
               | INPUT
               | SHOW_COLLECTION
               | SHOW_REACTION CHANGE_IMAGE
               | VIDEO 'WITH' SHOW_REACTION;

QUESTIONING : QUESTION 'OK->' SHOW_REACTION COLLECTING 'NOK->' SHOW_REACTION COLLECTING;

COLLECTING : | COLLECT | NOT_COLLECT;
```

Listing C.3: UBI, The Guardian Dragon application description file.

```
GAME={
{ START: STOP -> IMAGE120 FOREVER ,
START: STOP -> IMAGE108 MOUSE_CLICK } ,
POI101: STOP -> QUESTION101 OK-> IMAGE110 COLLECT NOK-> IMAGE116 MOUSE_CLICK ,
POI101: WAVE -> IMAGE109 DURATION=3 ,
POI104: STOP -> QUESTION102 OK-> IMAGE112 COLLECT NOK-> IMAGE116 MOUSE_CLICK ,
POI104: WAVE -> IMAGE111 DURATION=3 ,
POI103: STOP -> QUESTION103 OK-> IMAGE113 COLLECT NOK-> IMAGE116 MOUSE_CLICK ,
POI103: WAVE -> IMAGE101 DURATION=3 ,
POI102: STOP -> QUESTION104 OK-> IMAGE115 COLLECT NOK-> IMAGE116 MOUSE_CLICK ,
POI102: WAVE -> IMAGE114 DURATION=3 ,
{ POI105: MOVING_TO -> VIDEO101 WITH IMAGE105 ,
POI105: MOVING_TO -> VIDEO101 WITH IMAGE104 } ,
POI105: MOVING_AWAY -> VIDEO101 WITH IMAGE103 ,
POI106: WAVE -> IMAGE119 MOUSE_CLICK ,
POI106: POKE -> VIDEO101 WITH IMAGE103 ,
{ POI106: STOP -> VIDEO101 WITH IMAGE106 ,
POI106: STOP -> VIDEO101 WITH IMAGE107 } }, POI105: STOP -> IMAGE117 FOREVER
```

C.2 MagicLight

Listing C.4: MagicLight application space description.

```
Space Name
Art Gallery
Description
Art Gallery
Number of Roomsw
1
Room1
Name
MagicLight Room
Description
Bringing Deamons to Life.
Width
5
Height
4
Length
5
Cameras
CAMERA1,0,Webcam,0.0,0.0,1.0,43.0,57.0
CAMERA2,1,Kinect,2.5,0.0,1.0,65.0,80.0
POIs
NO_POI,No Point Of Interest,NO POI,,1.5,5.0,1.5
START,Start,START,,1.5,5.0,1.5
POI201,DOG,DOG,,1.5,5.0,1.5
POI202,SUNSET,SUNSET,,2.5,5.0,1.5
POI203,BEACH,BEACH,,3.5,5.0,1.5
POI204,CAMERA,CAMERA,,2.5,0.0,1.0
Virtual Elements
,,,,,,,,,,,,
Actions
Wave
Raise Hand
Click
Shake
Out of View
Moving Away
Moving To
Stop
Texts
INFO201,SHOW_TEXT: Guess the frame!,SHOW_TEXT: Guess the frame!,,,false,false,,
INFO202,SHOW_TEXT: Which picture do you like the most?,SHOW_TEXT: Which picture do you like the most
     ?,,,false,false,,
INFO203,SHOW_TEXT: Let’s play a game!,SHOW_TEXT: Let’s play a game!,,,false,false,,
Multimedia
IMAGE201,Image,Background Image,Background Image,Start,background.png,false,false,,
IMAGE202,Image,Mask,Mask,Start,mask.png,false,false,,
IMAGE203,Image,Angry,Angry,,angry.jpg,false,false,,
IMAGE204,Image,Laughing,Laughing,,laughing.jpg,false,false,,
IMAGE205,Image,Crying,Crying,,crying.jpg,false,false,,
IMAGE206,Image,Dizzy,Dizzy,,dizzy.jpg,false,false,,
IMAGE207,Image,Relax,Relax,,relax.jpg,false,false,,
IMAGE208,Image,Sleeping,Sleeping,,sleeping.jpg,false,false,,
IMAGE209,Image,Confused,Confused,,confused.jpg,false,false,,
IMAGE210,Image,Neutral,Neutral,,neutral.jpg,false,false,,
IMAGE211,Image,Mask,Which picture do you like the most?,Start,pictures.png,false,false,,
Questions
,,,,,,,
Requests
INPUT201,Which picture do you like the most? a)BEACH b)DOG c)SUNSET,pictures.jpg,DESTINATION,,,,Start
```

Listing C.5: MagicLight application grammar.

```
%lex
```

```
%%
\s+ {/* skip whitespace */}

[0-9]+("."[0-9]+)?\b {return 'NUMBER';}


/* UNIVERSAL POINTS OF INTEREST */
"NO_POI" {return 'POI';}
"START" {return 'POI';}

/* MAGICLIGHT - POINTS OF INTEREST */
"POI201" {return 'POI';}
"POI202" {return 'POI';}
"POI203" {return 'POI';}
"POI204" {return 'POI';}
"POI205" {return 'POI';}
"POI206" {return 'VIRTUAL_CHARACTER';}

/* TIME */
"WAIT" {return 'WAIT';}
"DURATION" {return 'DURATION';}
"NONE" {return 'NONE';}
"FOREVER" {return 'FOREVER';}
"MOUSE_CLICK" {return 'MOUSE_CLICK';}

/* BODY GESTURES */
"WAVE" {return 'BODY_ACTION';}
"RAISE_HAND" {return 'BODY_ACTION';}
"POKE" {return 'BODY_ACTION';}
"CLICK" {return 'BODY_ACTION';}
"TAP" {return 'BODY_ACTION';}

/* DEVICE GESTURES */
"SHAKE" {return 'DEVICE_ACTION';}
"TURN_DOWN" {return 'DEVICE_ACTION';}
"TURN_UP" {return 'DEVICE_ACTION';}
"POINT" {return 'POINT';}

/* MOVEMENT ACTIONS */
"STOP" {return 'MOVEMENT_ACTION';}
"OUT_OF_VIEW" {return 'MOVEMENT_ACTION';}
"MOVING_AWAY" {return 'MOVEMENT_ACTION';}
"MOVING_TO" {return 'MOVEMENT_ACTION';}

/* GAMEPLAY */
"DESTINATION" {return 'GAMEPLAY';}

/* GENERIC REACTIONS */
"CHANGE_IMAGE" {return 'CHANGE_IMAGE';}
"VIBRATE" {return 'SHOW_REACTION';}

/* MAGICLIGHT - REACTIONS */
"INFO201" {return 'SHOW_REACTION';}
"INFO202" {return 'SHOW_REACTION';}
"INFO203" {return 'SHOW_REACTION';}
"IMAGE201" {return 'SHOW_REACTION';}
"IMAGE202" {return 'SHOW_REACTION';}
"IMAGE203" {return 'SHOW_REACTION';}
"IMAGE204" {return 'SHOW_REACTION';}
"IMAGE205" {return 'SHOW_REACTION';}
"IMAGE206" {return 'SHOW_REACTION';}
"IMAGE207" {return 'SHOW_REACTION';}
"IMAGE208" {return 'SHOW_REACTION';}
"IMAGE209" {return 'SHOW_REACTION';}
"IMAGE210" {return 'SHOW_REACTION';}
"IMAGE211" {return 'SHOW_REACTION';}

/* INPUT */
```

```
"INPUT201" {return 'INPUT';}

/* COLLECTING */
"COLLECT" {return 'COLLECT';}
"DO_NOT_COLLECT" {return 'NOT_COLLECT';}
"SHOW_COLLECTION" {return 'SHOW_COLLECTION';}

"->" {return '->';}
"OK->" {return 'OK->';}
"NOK->" {return 'NOK->';}
"=" {return '=';}
":" {return ':';}
"{" {return '{';}
"}" {return '}';}
"," {return ',';}
"AND" {return 'AND';}
"OR" {return 'OR';}
"IFPROJECTOR" {return 'IFPROJECTOR';}
"IFNOPROJECTOR" {return 'IFNOPROJECTOR';}

/lex

/* operator associations and precedence */

%nonassoc '{' '}'
%right '='
%left ':'
%left '->'
%left 'AND' 'OR'
%right ','
%left '|'

%start SPACE_EXPLORATION

%% /* language grammar */

SPACE_EXPLORATION : GAME
                              | INTERACTIVE_TOUR
                              | TOUR;

GAME : '{' INTERACTION_SEQUENCE '}' ',' FINISH
         | IFPROJECTOR '{' '{' INTERACTION_SEQUENCE '}' ',' FINISH '}'
          IFNOPROJECTOR '{' '{' INTERACTION_SEQUENCE '}' ',' FINISH '}';

INTERACTIVE_TOUR : '{' INTERACTION_SEQUENCE '}'
                           | IFPROJECTOR '{' '{' INTERACTION_SEQUENCE '}' '}'
                            IFNOPROJECTOR '{' '{' INTERACTION_SEQUENCE '}' '}';

TOUR : '{' POI_SEQUENCE '}'
        | IFPROJECTOR '{' '{' POI_SEQUENCE '}' '}'
         IFNOPROJECTOR '{' '{' POI_SEQUENCE '}' '}';

FINISH : INTERACTION_SEQUENCE ;

VIRTUAL_LOCATION : VIRTUAL_CHARACTER | VIRTUAL_OBJECT ;

POI_SEQUENCE : POI
                       | POI_SEQUENCE ',' POI_SEQUENCE ;

INTERACTION_SEQUENCE : POI ':' ACTION TIME '->' REACTION TIME
                | POI ':' ACTION GAMEPLAY TIME '->' REACTION TIME
                | POI ':' '{' ACTION_SET '}' TIME '->' REACTION TIME
                | VIRTUAL_LOCATION ':' ACTION TIME '->' REACTION TIME
                | VIRTUAL_LOCATION ':' '{' ACTION_SET '}' TIME '->' REACTION TIME
                | '{' INTERACTION_SEQUENCE ',' INTERACTION_SEQUENCE '}'
                | '{' INTERACTION_SET ',' INTERACTION_SEQUENCE '}'
                | INTERACTION_SEQUENCE ',' INTERACTION_SEQUENCE;

INTERACTION_SET : INTERACTION_SEQUENCE 'OR' INTERACTION_SEQUENCE
```

```
                           | INTERACTION_SEQUENCE 'AND' INTERACTION_SEQUENCE;

TIME :
    | NONE
    | MOUSE_CLICK
    | FOREVER
    | DURATION '=' NUMBER
    | WAIT '=' NUMBER
    | DURATION '=' NUMBER WAIT '=' NUMBER;

ACTION : BODY_ACTION
           | MOVEMENT_ACTION
           | DEVICE_ACTION;

ACTION_SET : MOVEMENT_ACTION ',' BODY_ACTION
                  | MOVEMENT_ACTION ',' DEVICE_ACTION
                  | POINT ',' DEVICE_ACTION
                  | BODY_ACTION ',' DEVICE_ACTION;

REACTION : SHOW_REACTION
               | SHOW_REACTION COLLECTING
               | QUESTIONING
               | INPUT
               | SHOW_COLLECTION
               | SHOW_REACTION CHANGE_IMAGE;

QUESTIONING : QUESTION ANSWER
                     'OK->' SHOW_REACTION COLLECTING
                     'NOK->' SHOW_REACTION COLLECTING;

COLLECTING : | COLLECT | NOT_COLLECT;
```

Listing C.6: MagicLight application description file.

```
INTERACTIVE_TOUR={
{ START: STOP -> INFO203 DURATION=3 ,
START: STOP -> INFO202 MOUSE_CLICK ,
START: STOP -> INPUT201 ,
START: STOP -> INFO201 MOUSE_CLICK ,
START: STOP -> IMAGE201 FOREVER ,
START: STOP -> IMAGE202 FOREVER } ,
{ POI204: STOP DURATION=3 -> IMAGE207 ,
POI204: STOP DURATION=3 -> IMAGE208 } ,
POI205: MOVING_TO -> IMAGE209 ,
POI204: OUT_OF_VIEW -> IMAGE205 ,
{ POI204: SHAKE DURATION=3 -> IMAGE206 ,
POI204: SHAKE DURATION=3 -> IMAGE203 } ,
POI204: RAISE_HAND -> IMAGE204 ,
POI201: STOP DESTINATION -> IMAGE204 ,
POI201: MOVING_AWAY DESTINATION -> IMAGE210 ,
POI201: MOVING_TO DESTINATION -> IMAGE210 ,
POI202: STOP DESTINATION -> IMAGE204 ,
POI202: MOVING_AWAY DESTINATION -> IMAGE210 ,
POI202: MOVING_TO DESTINATION -> IMAGE210 ,
POI203: STOP DESTINATION -> IMAGE204 ,
POI203: MOVING_AWAY DESTINATION -> IMAGE210 ,
POI203: MOVING_TO DESTINATION -> IMAGE210 }
```

C.3 Gone Fishing

Listing C.7: Gone Fishing application space description.

```
Space Name
Aquarium
Description
Live fish and other underwater creatures.
Number of Rooms
1
Room1
Name
Gone Fishing Room
Description
Exploring a virtual sea bed.
Width
5
Height
4
Length
5
Cameras
CAMERA1,0,Webcam,0.0,0.0,1.0,43.0,57.0
CAMERA2,1,Kinect,2.5,0.0,1.0,65.0,80.0
POIs
POI0,NO_POI,NO_POI,,,,
POI1,Start,START,,,,
Virtual Elements
POI301,Fish 1,Fish 1,IMAGE302,Character,0,0,0,200,200,center,-1,-1,-1
POI302,Fish 2,Fish 2,IMAGE303,Character,0,0,0,200,200,center,-1,-1,-1
POI303,Fish 3,Fish 3,IMAGE304,Character,0,0,0,200,200,center,-1,-1,-1
POI304,Coral,Coral,IMAGE305,Object,0,0,0,200,200,center,-1,-1,-1
POI305,Seaweed 1,IMAGE306,Seaweed 1,Object,0,0,0,200,200,center,-1,-1,-1
POI306,Seaweed 2,IMAGE307,Seaweed 2,Object,0,0,0,200,200,center,-1,-1,-1
POI307,Decoration 1,IMAGE308,Decoration 1,Object,0,0,0,200,200,center,-1,-1,-1
POI308,Decoration 2,IMAGE309,Decoration 2,Object,0,0,0,200,200,center,-1,-1,-1
Actions
Point
Shake
Turn Down
Stop
Texts
INFO301,SHOW_TEXT: Missed!,SHOW_TEXT: Missed!,,,false,false,,
INFO302,SHOW_TEXT: Yes!,SHOW_TEXT: Yes!,,,false,false,,
INFO303,SHOW_TEXT: Gone fishing!,SHOW_TEXT: Gone Fishing!,,,false,false,,
Multimedia
IMAGE301,Image,Aim,Aim,Start,Aim.png,false,false,,
IMAGE302,Image,Fish 1,Fish 1,,Fish1.png,true,false,IMAGE,Fish1.png
IMAGE303,Image,Fish 2,Fish 2,,Fish2.png,true,false,IMAGE,Fish2.png
IMAGE304,Image,Fish 3,Fish 3,,Fish3.png,true,false,IMAGE,Fish3.png
IMAGE305,Image,Coral,Coral,,Coral1.png,true,false,IMAGE,Coral1.png
IMAGE306,Image,Seaweed 1,Seaweed 1,,Seaweed1.png,true,false,IMAGE,Seaweed1.png
IMAGE307,Image,Seaweed 2,Seaweed 2,,Seaweed2.png,true,false,IMAGE,Seaweed2.png
IMAGE308,Image,Decoration 1,Decoration 1,,Decoration1.png,true,false,IMAGE,Decoration1.png
IMAGE309,Image,Decoration 2,Decoration 2,,Decoration2.png,true,false,IMAGE,Decoration2.png
IMAGE310,Image,Aquarium,Aquarium,,Aquarium.png,false,false,,
IMAGE311,Image,Background Image,Background Image,Start,Background.png,false,false,,
IMAGE312,Image,Mask,Mask,Start,mask.png,false,false,,
Questions
,,,,,,,
Requests
,,,,,,,
```

Listing C.8: Gone Fishing application grammar.

```
%lex
```

```
%%
\s+ {/* skip whitespace */}

[0-9]+("."[0-9]+)?\b {return 'NUMBER';}


/* UNIVERSAL POINTS OF INTEREST */
"NO_POI" {return 'POI';}
"START" {return 'POI';}

/* GONE FISHING - POINTS OF INTEREST */
"POI301" {return 'VIRTUAL_CHARACTER';}
"POI302" {return 'VIRTUAL_CHARACTER';}
"POI303" {return 'VIRTUAL_CHARACTER';}
"POI304" {return 'VIRTUAL_OBJECT';}
"POI305" {return 'VIRTUAL_OBJECT';}
"POI306" {return 'VIRTUAL_OBJECT';}
"POI307" {return 'VIRTUAL_OBJECT';}
"POI308" {return 'VIRTUAL_OBJECT';}

/* TIME */
"WAIT" {return 'WAIT';}
"DURATION" {return 'DURATION';}
"NONE" {return 'NONE';}
"FOREVER" {return 'FOREVER';}
"MOUSE_CLICK" {return 'MOUSE_CLICK';}

/* BODY GESTURES */
"WAVE" {return 'BODY_ACTION';}
"RAISE_HAND" {return 'BODY_ACTION';}
"POKE" {return 'BODY_ACTION';}
"CLICK" {return 'BODY_ACTION';}
"TAP" {return 'BODY_ACTION';}

/* DEVICE GESTURES */
"SHAKE" {return 'DEVICE_ACTION';}
"TURN_DOWN" {return 'DEVICE_ACTION';}
"TURN_UP" {return 'DEVICE_ACTION';}
"POINT" {return 'POINT';}

/* MOVEMENT ACTIONS */
"STOP" {return 'MOVEMENT_ACTION';}
"OUT_OF_VIEW" {return 'MOVEMENT_ACTION';}
"MOVING_AWAY" {return 'MOVEMENT_ACTION';}
"MOVING_TO" {return 'MOVEMENT_ACTION';}

/* GENERIC REACTIONS */
"CHANGE_IMAGE" {return 'CHANGE_IMAGE';}
"VIBRATE" {return 'SHOW_REACTION';}

/* GONE FISHING - REACTIONS */
"INFO301" {return 'SHOW_REACTION';}
"INFO302" {return 'SHOW_REACTION';}
"INFO303" {return 'SHOW_REACTION';}
"IMAGE301" {return 'SHOW_REACTION';}
"IMAGE302" {return 'SHOW_REACTION';}
"IMAGE303" {return 'SHOW_REACTION';}
"IMAGE304" {return 'SHOW_REACTION';}
"IMAGE305" {return 'SHOW_REACTION';}
"IMAGE306" {return 'SHOW_REACTION';}
"IMAGE307" {return 'SHOW_REACTION';}
"IMAGE308" {return 'SHOW_REACTION';}
"IMAGE309" {return 'SHOW_REACTION';}
"IMAGE310" {return 'SHOW_REACTION';}
"IMAGE311" {return 'SHOW_REACTION';}
"IMAGE312" {return 'SHOW_REACTION';}

/* COLLECTING */
"COLLECT" {return 'COLLECT';}
```

```
"DO_NOT_COLLECT" {return 'NOT_COLLECT';}
"SHOW_COLLECTION" {return 'SHOW_COLLECTION';}

"->" {return '->';}
"OK->" {return 'OK->';}
"NOK->" {return 'NOK->';}
"=" {return '=';}
":" {return ':';}
"{" {return '{';}
"}" {return '}';}
"," {return ',';}
"AND" {return 'AND';}
"OR" {return 'OR';}
"IFPROJECTOR" {return 'IFPROJECTOR';}
"IFNOPROJECTOR" {return 'IFNOPROJECTOR';}

/lex

/* operator associations and precedence */

%nonassoc '{' '}'
%right '='
%left ':'
%left '->'
%left 'AND' 'OR'
%right ','
%left '|'

%start SPACE_EXPLORATION

%% /* language grammar */

SPACE_EXPLORATION : GAME
                              | INTERACTIVE_TOUR
                              | TOUR;

GAME : '{' INTERACTION_SEQUENCE '}' ',' FINISH
         | IFPROJECTOR '{' '{' INTERACTION_SEQUENCE '}' ',' FINISH '}'
          IFNOPROJECTOR '{' '{' INTERACTION_SEQUENCE '}' ',' FINISH '}';

INTERACTIVE_TOUR : '{' INTERACTION_SEQUENCE '}'
                           | IFPROJECTOR '{' '{' INTERACTION_SEQUENCE '}' '}'
                            IFNOPROJECTOR '{' '{' INTERACTION_SEQUENCE '}' '}';

TOUR : '{' POI_SEQUENCE '}'
        | IFPROJECTOR '{' '{' POI_SEQUENCE '}' '}'
         IFNOPROJECTOR '{' '{' POI_SEQUENCE '}' '}';

FINISH : INTERACTION_SEQUENCE ;

VIRTUAL_LOCATION : VIRTUAL_CHARACTER | VIRTUAL_OBJECT ;

POI_SEQUENCE : POI
                      | POI_SEQUENCE ',' POI_SEQUENCE ;

INTERACTION_SEQUENCE : POI ':' ACTION TIME '->' REACTION TIME
                | POI ':' ACTION GAMEPLAY TIME '->' REACTION TIME
                | POI ':' '{' ACTION_SET '}' TIME '->' REACTION TIME
                | VIRTUAL_LOCATION ':' ACTION TIME '->' REACTION TIME
                | VIRTUAL_LOCATION ':' '{' ACTION_SET '}' TIME '->' REACTION TIME
                | '{' INTERACTION_SEQUENCE ',' INTERACTION_SEQUENCE '}'
                | '{' INTERACTION_SET ',' INTERACTION_SEQUENCE '}'
                | INTERACTION_SEQUENCE ',' INTERACTION_SEQUENCE;

INTERACTION_SET : INTERACTION_SEQUENCE 'OR' INTERACTION_SEQUENCE
                         | INTERACTION_SEQUENCE 'AND' INTERACTION_SEQUENCE;

TIME :
   | NONE
```

```
    | MOUSE_CLICK
    | FOREVER
    | DURATION '=' NUMBER
    | WAIT '=' NUMBER
    | DURATION '=' NUMBER WAIT '=' NUMBER;

ACTION : BODY_ACTION
           | MOVEMENT_ACTION
           | DEVICE_ACTION;

ACTION_SET : MOVEMENT_ACTION ',' BODY_ACTION
                  | MOVEMENT_ACTION ',' DEVICE_ACTION
                  | POINT ',' DEVICE_ACTION
                  | BODY_ACTION ',' DEVICE_ACTION;

REACTION : SHOW_REACTION
               | SHOW_REACTION COLLECTING
               | QUESTIONING
               | INPUT
               | SHOW_COLLECTION
               | SHOW_REACTION CHANGE_IMAGE;

QUESTIONING : QUESTION ANSWER
                     'OK->' SHOW_REACTION COLLECTING
                     'NOK->' SHOW_REACTION COLLECTING;

COLLECTING : | COLLECT | NOT_COLLECT;
```

Listing C.9: Gone Fishing application description file.

```
INTERACTIVE_TOUR={
{ START: STOP -> INFO303 DURATION=3 ,
START: STOP -> IMAGE311 FOREVER ,
START: STOP -> IMAGE301 FOREVER ,
START: STOP -> IMAGE312 FOREVER } ,
POI301: {POINT , SHAKE} -> INFO302 COLLECT DURATION=3 ,
POI302: {POINT , SHAKE} -> INFO302 COLLECT DURATION=3 ,
POI303: {POINT , SHAKE} -> INFO302 COLLECT DURATION=3 ,
POI304: {POINT , SHAKE} -> INFO302 COLLECT DURATION=3 ,
POI305: {POINT , SHAKE} -> INFO302 COLLECT DURATION=3 ,
POI306: {POINT , SHAKE} -> INFO302 COLLECT DURATION=3 ,
POI307: {POINT , SHAKE} -> INFO302 COLLECT DURATION=3 ,
POI308: {POINT , SHAKE} -> INFO302 COLLECT DURATION=3 ,
NO_POI: {POINT , SHAKE} -> INFO301 DURATION=3 ,
{ NO_POI: TURN_DOWN -> IMAGE310 FOREVER ,
NO_POI: STOP -> SHOW_COLLECTION FOREVER ,
NO_POI: TURN_DOWN -> IMAGE311 FOREVER ,
NO_POI: STOP -> IMAGE301 FOREVER } ,
NO_POI: STOP -> IMAGE312 FOREVER }
```

C.4 Haunted House

Listing C.10: Haunted House application space description.

```
Space Name
Haunted House
Description
A mansion with ghosts, vampires and other creapy creatures.
Number of Rooms
1
Room1
Name
Haunted House Room
Description
Haunted house.
Width
5
Height
4
Length
5
Cameras
CAMERA1,0,Webcam,0.0,0.0,1.0,43.0,57.0
CAMERA2,1,Kinect,2.5,0.0,1.0,65.0,80.0
POIs
POI0,NO_POI,NO_POI,,,,
POI1,Start,START,,,,
Virtual Elements
POI401,GHOST,GHOST,IMAGE403,Character,0,0,0,200,200,center,-1,-1,-1
POI402,OPEN_CHEST,OPEN_CHEST,IMAGE404,Object,0,0,0,200,200,center,-1,-1,-1
POI403,CLOSED_CHEST,CLOSED_CHEST,IMAGE405,Object,0,0,0,200,200,center,-1,-1,-1
POI404,MICE,MICE,IMAGE406,Character,0,0,0,200,200,center,-1,-1,-1
POI405,BATS,BATS,IMAGE407,Character,0,0,0,200,200,center,-1,-1,-1
POI406,PAINTING,PAINTING,IMAGE408,Object,0,0,0,200,200,center,-1,-1,-1
POI407,VAMPIRE,VAMPIRE,IMAGE409,Character,0,0,0,200,200,center,-1,-1,-1
Actions
Texts
INFO401,SHOW_TEXT: ??,SHOW_TEXT: ??,,,false,false,,
INFO402,SHOW_TEXT: BUUU!,SHOW_TEXT: BUUU!,,,false,false,,
INFO403,SHOW_TEXT: Haunted House!,SHOW_TEXT: Haunted House!,,,false,false,,
Multimedia
IMAGE401,Image,Aim,Aim,Start,aim.png,false,false,,
IMAGE402,Image,Ghost,Ghost,,ghost.jpg,false,false,,
IMAGE403,Image,Open Chest,Open Chest,,open_chest.png,false,false,,
IMAGE404,Image,Closed Chest,Closed Chest,,closed_chest.png,false,false,,
IMAGE405,Image,Mice,Mice,,mice.jpg,false,false,,
IMAGE406,Image,Bats,Bats,,bats.jpg,false,false,,
IMAGE407,Image,Painting,Painting,,painting.jpg,false,false,,
IMAGE408,Image,Vampire,Vampire,,vampire.jpg,false,false,,
IMAGE409,Image,Background Color,Background Color,Start,background.png,false,false,,
SOUND401,Sound,Scream 1,Scream 1,,scare1.wav,false,false,,
SOUND402,Sound,Scream 2,Scream 2,,scare2.wav,false,false,,
Questions
,,,,,,,
Requests
,,,,,,,
```

Listing C.11: Haunted House application grammar.

```
%lex

%%
\s+ {/* skip whitespace */}

[0-9]+("."[0-9]+)?\b {return 'NUMBER';}
```

```
/* UNIVERSAL POINTS OF INTEREST */
"NO_POI" {return 'POI';}
"START" {return 'POI';}

/* HAUNTED HOUSE - POINTS OF INTEREST */
"POI401" {return 'VIRTUAL_OBJECT';}
"POI402" {return 'VIRTUAL_OBJECT';}
"POI403" {return 'VIRTUAL_OBJECT';}
"POI404" {return 'VIRTUAL_OBJECT';}
"POI405" {return 'VIRTUAL_OBJECT';}
"POI406" {return 'VIRTUAL_OBJECT';}
"POI407" {return 'VIRTUAL_OBJECT';}

/* TIME */
"WAIT" {return 'WAIT';}
"DURATION" {return 'DURATION';}
"NONE" {return 'NONE';}
"FOREVER" {return 'FOREVER';}
"MOUSE_CLICK" {return 'MOUSE_CLICK';}

/* BODY GESTURES */
"WAVE" {return 'BODY_ACTION';}
"RAISE_HAND" {return 'BODY_ACTION';}
"POKE" {return 'BODY_ACTION';}
"CLICK" {return 'BODY_ACTION';}
"TAP" {return 'BODY_ACTION';}

/* DEVICE GESTURES */
"SHAKE" {return 'DEVICE_ACTION';}
"TURN_DOWN" {return 'DEVICE_ACTION';}
"TURN_UP" {return 'DEVICE_ACTION';}
"POINT" {return 'POINT';}

/* MOVEMENT ACTIONS */
"STOP" {return 'MOVEMENT_ACTION';}
"OUT_OF_VIEW" {return 'MOVEMENT_ACTION';}
"MOVING_AWAY" {return 'MOVEMENT_ACTION';}
"MOVING_TO" {return 'MOVEMENT_ACTION';}

/* GENERIC REACTIONS */
"CHANGE_IMAGE" {return 'CHANGE_IMAGE';}
"VIBRATE" {return 'SHOW_REACTION';}

/* HAUNTED HOUSE - REACTIONS */
"INFO401" {return 'SHOW_REACTION';}
"INFO402" {return 'SHOW_REACTION';}
"INFO403" {return 'SHOW_REACTION';}
"IMAGE401" {return 'SHOW_REACTION';}
"IMAGE402" {return 'SHOW_REACTION';}
"IMAGE403" {return 'SHOW_REACTION';}
"IMAGE404" {return 'SHOW_REACTION';}
"IMAGE405" {return 'SHOW_REACTION';}
"IMAGE406" {return 'SHOW_REACTION';}
"IMAGE407" {return 'SHOW_REACTION';}
"IMAGE408" {return 'SHOW_REACTION';}
"IMAGE409" {return 'SHOW_REACTION';}
"SOUND401" {return 'SHOW_REACTION';}
"SOUND402" {return 'SHOW_REACTION';}


/* COLLECTING */
"COLLECT" {return 'COLLECT';}
"DO_NOT_COLLECT" {return 'NOT_COLLECT';}
"SHOW_COLLECTION" {return 'SHOW_COLLECTION';}

"->" {return '->';}
"OK->" {return 'OK->';}
"NOK->" {return 'NOK->';}
"=" {return '=';}
```

```
":" {return ':';}
"{" {return '{';}
"}" {return '}';}
"," {return ',';}
"AND" {return 'AND';}
"OR" {return 'OR';}
"IFPROJECTOR" {return 'IFPROJECTOR';}
"IFNOPROJECTOR" {return 'IFNOPROJECTOR';}

/lex

/* operator associations and precedence */

%nonassoc '{' '}'
%right '='
%left ':'
%left '->'
%left 'AND' 'OR'
%right ','
%left '|'

%start SPACE_EXPLORATION

%% /* language grammar */

SPACE_EXPLORATION : GAME
                              | INTERACTIVE_TOUR
                              | TOUR;

GAME : '{' INTERACTION_SEQUENCE '}' ',' FINISH
         | IFPROJECTOR '{' '{' INTERACTION_SEQUENCE '}' ',' FINISH '}'
          IFNOPROJECTOR '{' '{' INTERACTION_SEQUENCE '}' ',' FINISH '}';

INTERACTIVE_TOUR : '{' INTERACTION_SEQUENCE '}'
                           | IFPROJECTOR '{' '{' INTERACTION_SEQUENCE '}' '}'
                            IFNOPROJECTOR '{' '{' INTERACTION_SEQUENCE '}' '}';

TOUR : '{' POI_SEQUENCE '}'
        | IFPROJECTOR '{' '{' POI_SEQUENCE '}' '}'
         IFNOPROJECTOR '{' '{' POI_SEQUENCE '}' '}';

FINISH : INTERACTION_SEQUENCE ;

VIRTUAL_LOCATION : VIRTUAL_CHARACTER | VIRTUAL_OBJECT ;

POI_SEQUENCE : POI
                      | POI_SEQUENCE ',' POI_SEQUENCE ;

INTERACTION_SEQUENCE : POI ':' ACTION TIME '->' REACTION TIME
                | POI ':' ACTION GAMEPLAY TIME '->' REACTION TIME
                | POI ':' '{' ACTION_SET '}' TIME '->' REACTION TIME
                | VIRTUAL_LOCATION ':' ACTION TIME '->' REACTION TIME
                | VIRTUAL_LOCATION ':' '{' ACTION_SET '}' TIME '->' REACTION TIME
                | '{' INTERACTION_SEQUENCE ',' INTERACTION_SEQUENCE '}'
                | '{' INTERACTION_SET ',' INTERACTION_SEQUENCE '}'
                | INTERACTION_SEQUENCE ',' INTERACTION_SEQUENCE;

INTERACTION_SET : INTERACTION_SEQUENCE 'OR' INTERACTION_SEQUENCE
                          | INTERACTION_SEQUENCE 'AND' INTERACTION_SEQUENCE;

TIME :
    | NONE
    | MOUSE_CLICK
    | FOREVER
    | DURATION '=' NUMBER
    | WAIT '=' NUMBER
    | DURATION '=' NUMBER WAIT '=' NUMBER;

ACTION : BODY_ACTION
```

```
          | MOVEMENT_ACTION
          | DEVICE_ACTION;

ACTION_SET : MOVEMENT_ACTION ',' BODY_ACTION
                | MOVEMENT_ACTION ',' DEVICE_ACTION
                | POINT ',' DEVICE_ACTION
                | BODY_ACTION ',' DEVICE_ACTION;

REACTION : SHOW_REACTION
             | SHOW_REACTION COLLECTING
             | QUESTIONING
             | INPUT
             | SHOW_COLLECTION
             | SHOW_REACTION CHANGE_IMAGE;

QUESTIONING : QUESTION ANSWER
                  'OK->' SHOW_REACTION COLLECTING
                  'NOK->' SHOW_REACTION COLLECTING;

COLLECTING : | COLLECT | NOT_COLLECT;
```

Listing C.12: Haunted House application description file.

```
INTERACTIVE_TOUR={
{ START: STOP -> INFO403 MOUSE_CLICK ,
START: STOP -> IMAGE409 FOREVER ,
START: STOP -> IMAGE401 FOREVER } ,
{ POI402: {POINT , SHAKE} -> INFO402 ,
POI402: STOP -> IMAGE403 CHANGE_IMAGE ,
POI402: STOP -> IMAGE402 DURATION=3 ,
POI402: STOP -> SOUND401 ,
POI402: STOP -> IMAGE404 CHANGE_IMAGE } ,
{ POI405: {POINT , SHAKE} -> SOUND402 ,
POI405: STOP -> INFO402 ,
POI405: STOP -> IMAGE408 DURATION=3 } ,
POI403: {POINT , SHAKE} -> INFO402 DURATION=3 ,
POI404: {POINT , SHAKE} -> INFO402 DURATION=3 ,
POI407: {POINT , SHAKE} -> SOUND401 ,
POI406: {POINT , SHAKE} -> SOUND402 ,
NO_POI: SHAKE -> INFO401 }
```

C.5 Toy Exhibition

Listing C.13: The Toy Exhibition space description.

```
Space Name
Toy Exhibition
Description
Sharing old and new toys.
Number of Rooms
1
Room1
Name
Toys
Description
Toys
Width
5
Height
4
Length
5
Cameras
CAMERA1,0,Webcam,0.0,0.0,1.0,43.0,57.0
CAMERA2,1,Kinect,2.5,0.0,1.0,65.0,80.0
POIs
POI11,Toys,TOY,,1.5,0.5,0.5
POI12,Puzzles,PUZZLES,,2.5,0.5,0.5
POI13,Musical Instruments,MUSICAL_INSTRUMENTS,,3.5,0.5,0.5
Virtual Elements
,,,,,,,,,,,,,
Actions
Wave
Raise Hand
Click
Texts
INFO1,Hello!,Hello!,,,false,false,,
INFO2,See you soon!,See you soon!,,,false,false,,
INFO3,How are you?,How are you?,,,false,false,,
Multimedia
IMAGE1,Image,Star,Star,,star.png,false,false,,
IMAGE2,Image,Sword,Sword,,sword.png,false,false,,
Questions
,,,,,,,
Requests
,,,,,,,
```

Listing C.14: A Toy Exhibition application grammar.

```
%lex

%%
\s+ {/* skip whitespace */}

[0-9]+("."[0-9]+)?\b {return 'NUMBER';}


/* UNIVERSAL POINTS OF INTEREST */
"NO_POI" {return 'POI';}
"START" {return 'POI';}

/* URSPACE - POINTS OF INTEREST */
"POI11" {return 'POI';}
"POI12" {return 'POI';}
"POI13" {return 'POI';}

/* TIME */
"WAIT" {return 'WAIT';}
```

```
"DURATION" {return 'DURATION';}
"NONE" {return 'NONE';}
"FOREVER" {return 'FOREVER';}
"MOUSE_CLICK" {return 'MOUSE_CLICK';}

/* BODY GESTURES */
"WAVE" {return 'BODY_ACTION';}
"RAISE_HAND" {return 'BODY_ACTION';}
"POKE" {return 'BODY_ACTION';}
"CLICK" {return 'BODY_ACTION';}
"TAP" {return 'BODY_ACTION';}

/* DEVICE GESTURES */
"SHAKE" {return 'DEVICE_ACTION';}
"TURN_DOWN" {return 'DEVICE_ACTION';}
"TURN_UP" {return 'DEVICE_ACTION';}
"POINT" {return 'POINT';}

/* MOVEMENT ACTIONS */
"STOP" {return 'MOVEMENT_ACTION';}
"OUT_OF_VIEW" {return 'MOVEMENT_ACTION';}
"MOVING_AWAY" {return 'MOVEMENT_ACTION';}
"MOVING_TO" {return 'MOVEMENT_ACTION';}

/* GENERIC REACTIONS */
"CHANGE_IMAGE" {return 'CHANGE_IMAGE';}
"VIBRATE" {return 'SHOW_REACTION';}

/* URSPACE - REACTIONS */
"IMAGE1" {return 'SHOW_REACTION';}
"IMAGE2" {return 'SHOW_REACTION';}
"INFO1" {return 'SHOW_REACTION';}
"INFO2" {return 'SHOW_REACTION';}
"INFO3" {return 'SHOW_REACTION';}
"SOUND1" {return 'SHOW_REACTION';}

/* COLLECTING */
"COLLECT" {return 'COLLECT';}
"DO_NOT_COLLECT" {return 'NOT_COLLECT';}
"SHOW_COLLECTION" {return 'SHOW_COLLECTION';}

"->" {return '->';}
"OK->" {return 'OK->';}
"NOK->" {return 'NOK->';}
"=" {return '=';}
":" {return ':';}
"{" {return '{';}
"}" {return '}';}
"," {return ',';}
"AND" {return 'AND';}
"OR" {return 'OR';}
"IFPROJECTOR" {return 'IFPROJECTOR';}
"IFNOPROJECTOR" {return 'IFNOPROJECTOR';}

/lex

/* operator associations and precedence */

%nonassoc '{' '}'
%right '='
%left ':'
%left '->'
%left 'AND' 'OR'
%right ','
%left '|'

%start SPACE_EXPLORATION

%% /* language grammar */
```

```
SPACE_EXPLORATION : GAME
                               | INTERACTIVE_TOUR
                               | TOUR;

GAME : '{' INTERACTION_SEQUENCE '}' ',' FINISH
         | IFPROJECTOR '{' '{' INTERACTION_SEQUENCE '}' ',' FINISH '}'
          IFNOPROJECTOR '{' '{' INTERACTION_SEQUENCE '}' ',' FINISH '}';

INTERACTIVE_TOUR : '{' INTERACTION_SEQUENCE '}'
                            | IFPROJECTOR '{' '{' INTERACTION_SEQUENCE '}' '}'
                             IFNOPROJECTOR '{' '{' INTERACTION_SEQUENCE '}' '}';

TOUR : '{' POI_SEQUENCE '}'
        | IFPROJECTOR '{' '{' POI_SEQUENCE '}' '}'
         IFNOPROJECTOR '{' '{' POI_SEQUENCE '}' '}';

FINISH : INTERACTION_SEQUENCE ;

VIRTUAL_LOCATION : VIRTUAL_CHARACTER | VIRTUAL_OBJECT ;

POI_SEQUENCE : POI
                       | POI_SEQUENCE ',' POI_SEQUENCE ;

INTERACTION_SEQUENCE : POI ':' ACTION TIME '->' REACTION TIME
                | POI ':' ACTION GAMEPLAY TIME '->' REACTION TIME
                | POI ':' '{' ACTION_SET '}' TIME '->' REACTION TIME
                | VIRTUAL_LOCATION ':' ACTION TIME '->' REACTION TIME
                | VIRTUAL_LOCATION ':' '{' ACTION_SET '}' TIME '->' REACTION TIME
                | '{' INTERACTION_SEQUENCE ',' INTERACTION_SEQUENCE '}'
                | '{' INTERACTION_SET ',' INTERACTION_SEQUENCE '}'
                | INTERACTION_SEQUENCE ',' INTERACTION_SEQUENCE;

INTERACTION_SET : INTERACTION_SEQUENCE 'OR' INTERACTION_SEQUENCE
                          | INTERACTION_SEQUENCE 'AND' INTERACTION_SEQUENCE;

TIME :
    | NONE
    | MOUSE_CLICK
    | FOREVER
    | DURATION '=' NUMBER
    | WAIT '=' NUMBER
    | DURATION '=' NUMBER WAIT '=' NUMBER;

ACTION : BODY_ACTION
           | MOVEMENT_ACTION
           | DEVICE_ACTION;

ACTION_SET : MOVEMENT_ACTION ',' BODY_ACTION
                  | MOVEMENT_ACTION ',' DEVICE_ACTION
                  | POINT ',' DEVICE_ACTION
                  | BODY_ACTION ',' DEVICE_ACTION;

REACTION : SHOW_REACTION
               | SHOW_REACTION COLLECTING
               | QUESTIONING
               | INPUT
               | SHOW_COLLECTION
               | SHOW_REACTION CHANGE_IMAGE;

QUESTIONING : QUESTION ANSWER
                     'OK->' SHOW_REACTION COLLECTING
                     'NOK->' SHOW_REACTION COLLECTING;

COLLECTING : COLLECT | NOT_COLLECT;
```

Listing C.15: Toy Exhibition application description file example.

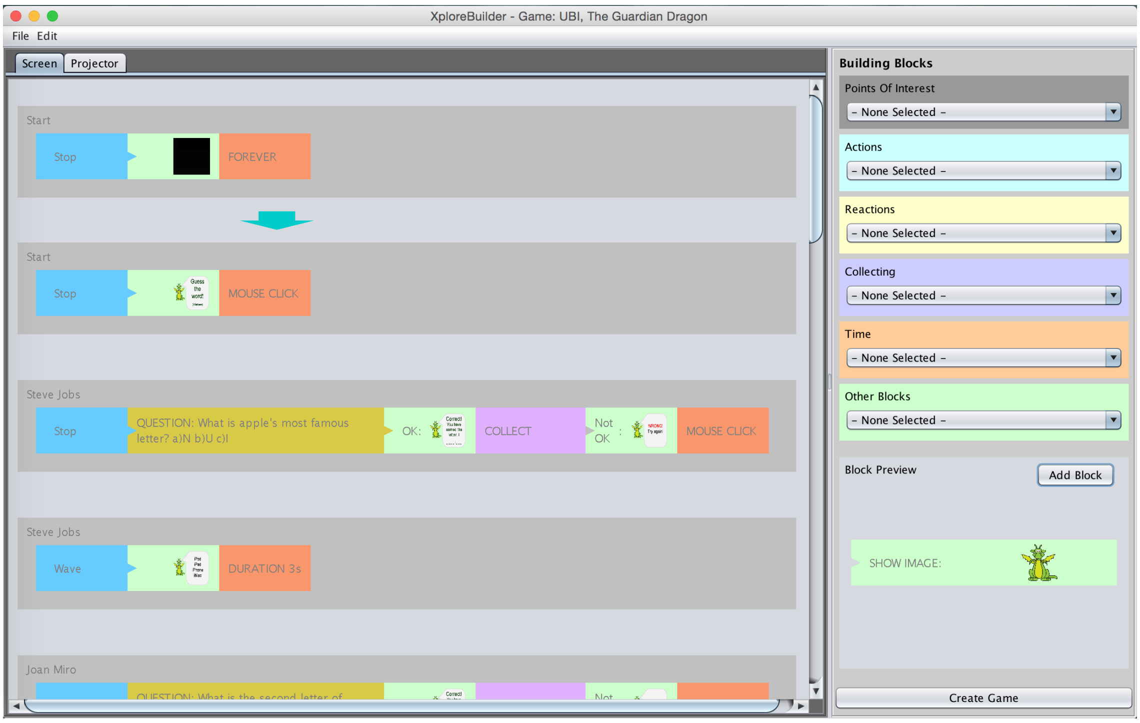
```
GAME={
POI13: RAISE_HAND -> VIBRATE ,
POI11: CLICK -> IMAGE2 ,
POI11: WAVE -> INFO1 },
POI12: CLICK -> INFO2
```

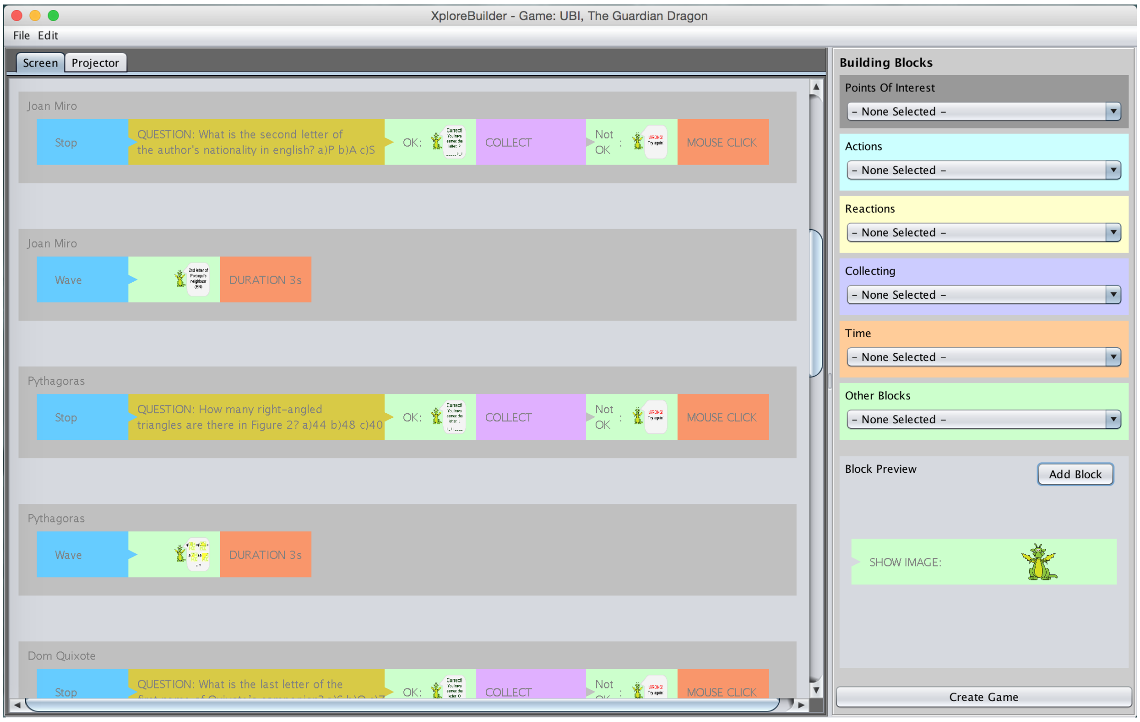
Appendix D

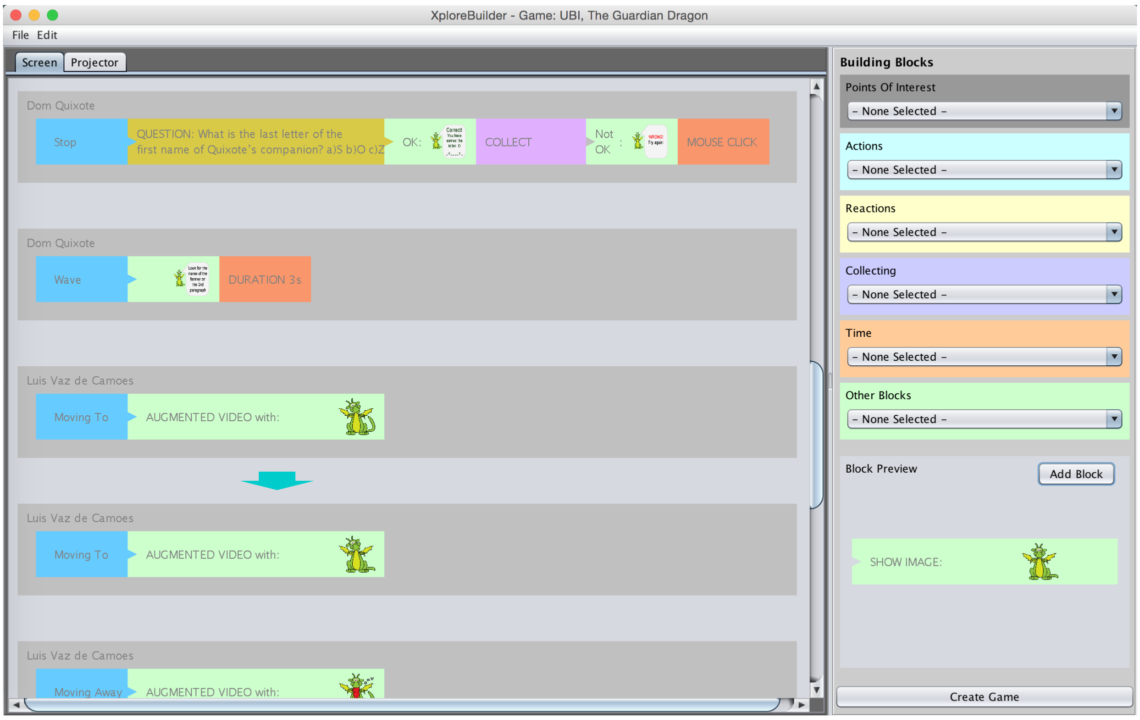
Application Design Examples

Five case studies where the XploreBuilder application was used are presented in the next sections. These case studies correspond to the applications developed in the scope of this research and that are described in Section 4. For each case study the design of the interactions is presented.

D.1 UBI, The Guardian Dragon




XploreBuilder - Game: UBI, The Guardian Dragon
File Edit
Screen Projector
Joan Miro
Stop
QUESTION: What is the second letter of the author's nationality in english? a)P b)A c)S
OK:
COLLECT
Not OK
MOUSE CLICK
Joan Miro
Wave
DURATION 3s
Pythagoras
Stop
QUESTION: How many right-angled triangles are there in Figure 2? a)44 b)48 c)40
OK:
COLLECT
Not OK
MOUSE CLICK
Pythagoras
Wave
DURATION 3s
Dom Quixote
Stop
QUESTION: What is the last letter of the
OK:
COLLECT
Not
MOUSE CLICK
Building Blocks
Points Of Interest
– None Selected –
Actions
– None Selected –
Reactions
– None Selected –
Collecting
– None Selected –
Time
– None Selected –
Other Blocks
– None Selected –
Block Preview
Add Block
SHOW IMAGE:
Create Game


XploreBuilder - Game: UBI, The Guardian Dragon
File Edit
Screen Projector
Dom Quixote
Stop
QUESTION: What is the last letter of the first name of Quixote's companion? a)S b)O c)Z
OK:
COLLECT
Not OK
MOUSE CLICK
Dom Quixote
Wave
DURATION 3s
Luis Vaz de Camoes
Moving To
AUGMENTED VIDEO with:
Luis Vaz de Camoes
Moving To
AUGMENTED VIDEO with:
Luis Vaz de Camoes
Moving Away
AUGMENTED VIDEO with:
Building Blocks
Points Of Interest
– None Selected –
Actions
– None Selected –
Reactions
– None Selected –
Collecting
– None Selected –
Time
– None Selected –
Other Blocks
– None Selected –
Block Preview
Add Block
SHOW IMAGE:
Create Game

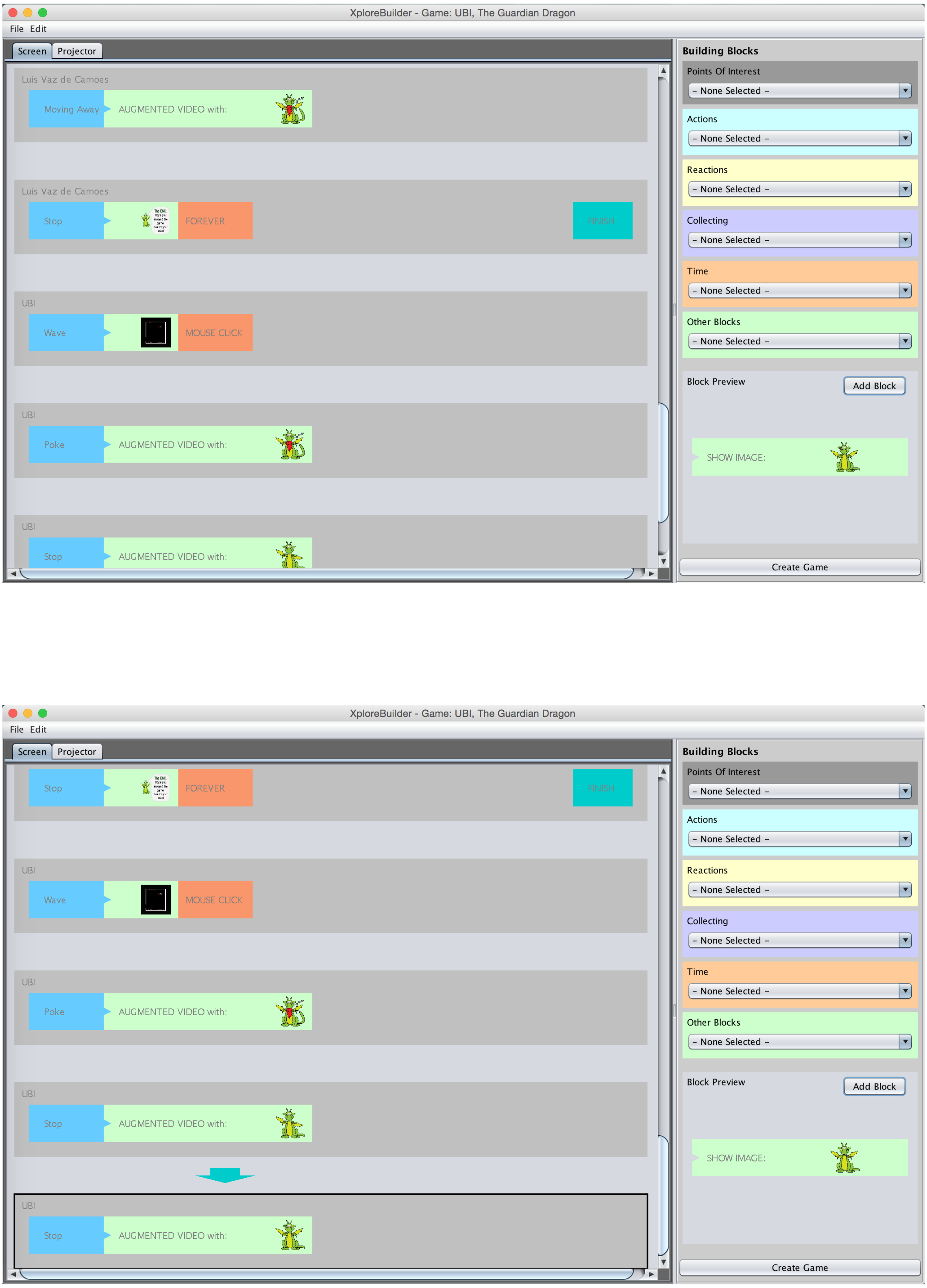
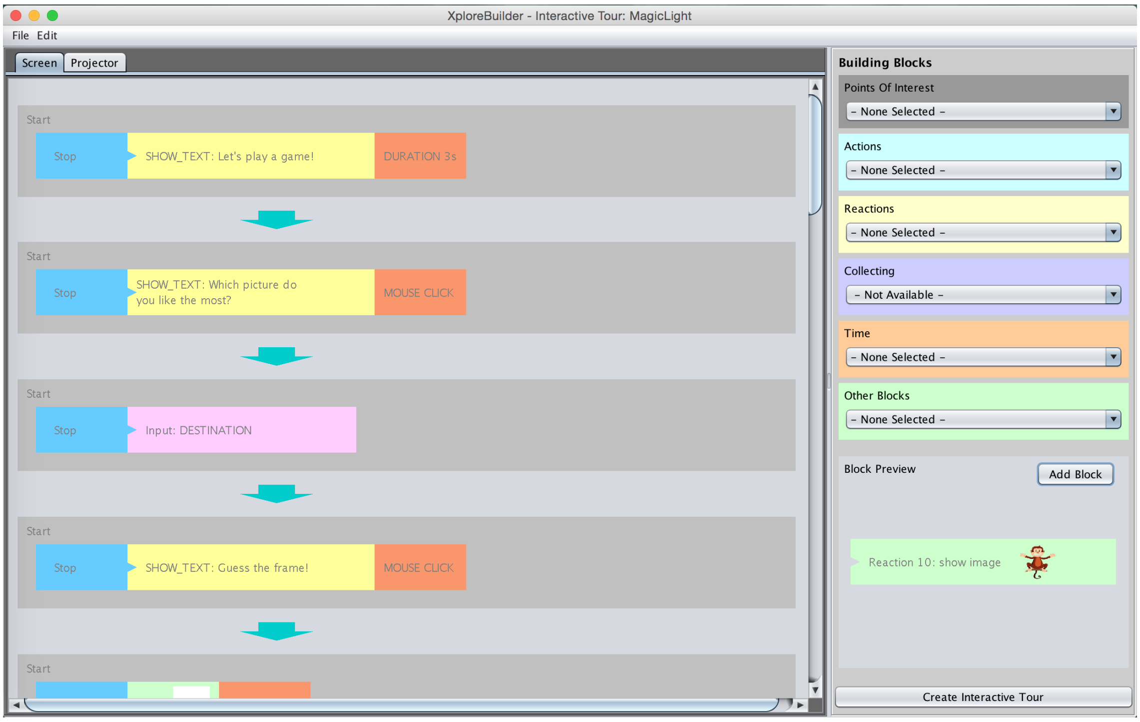
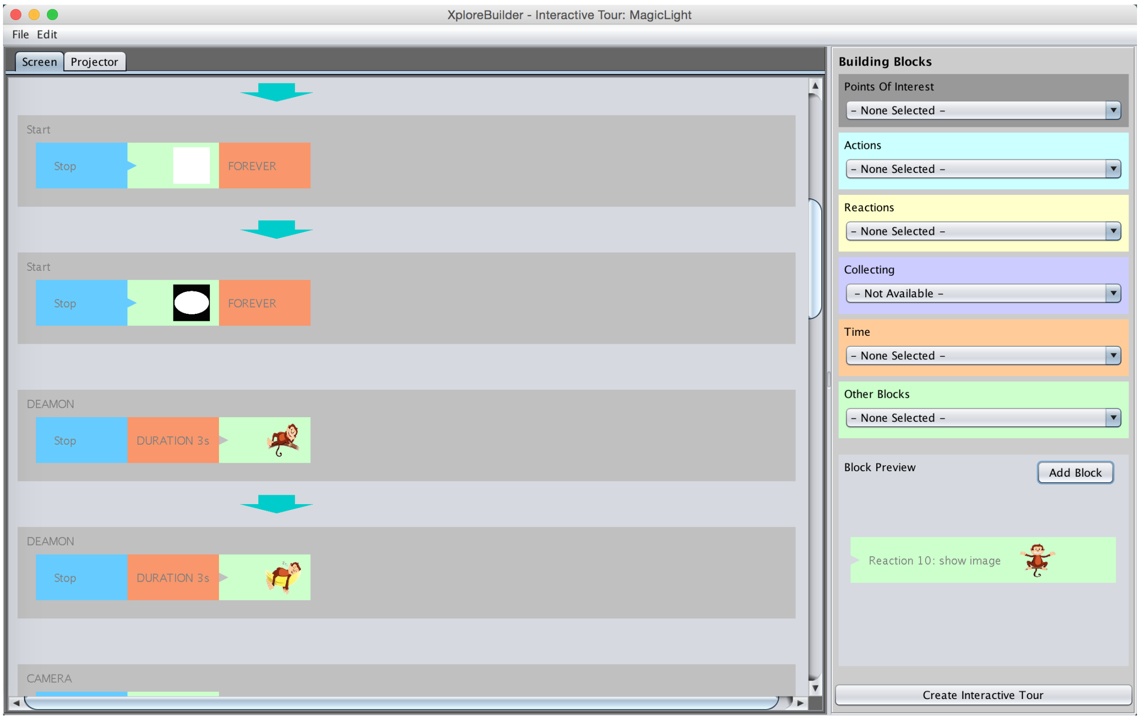
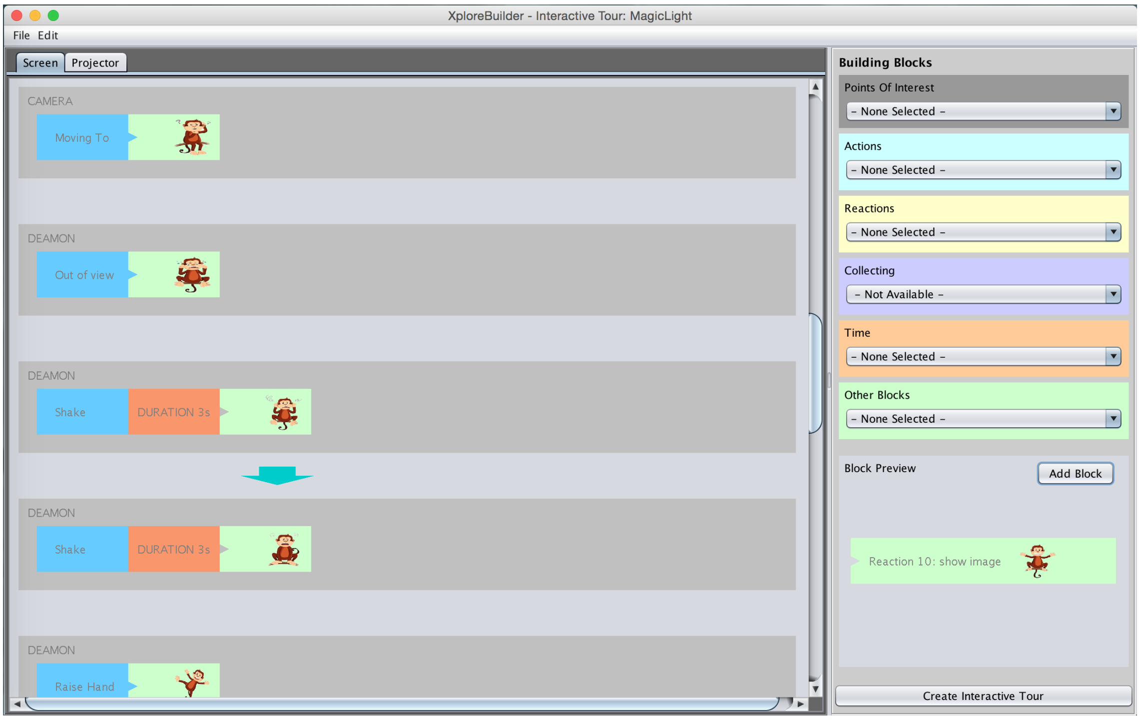


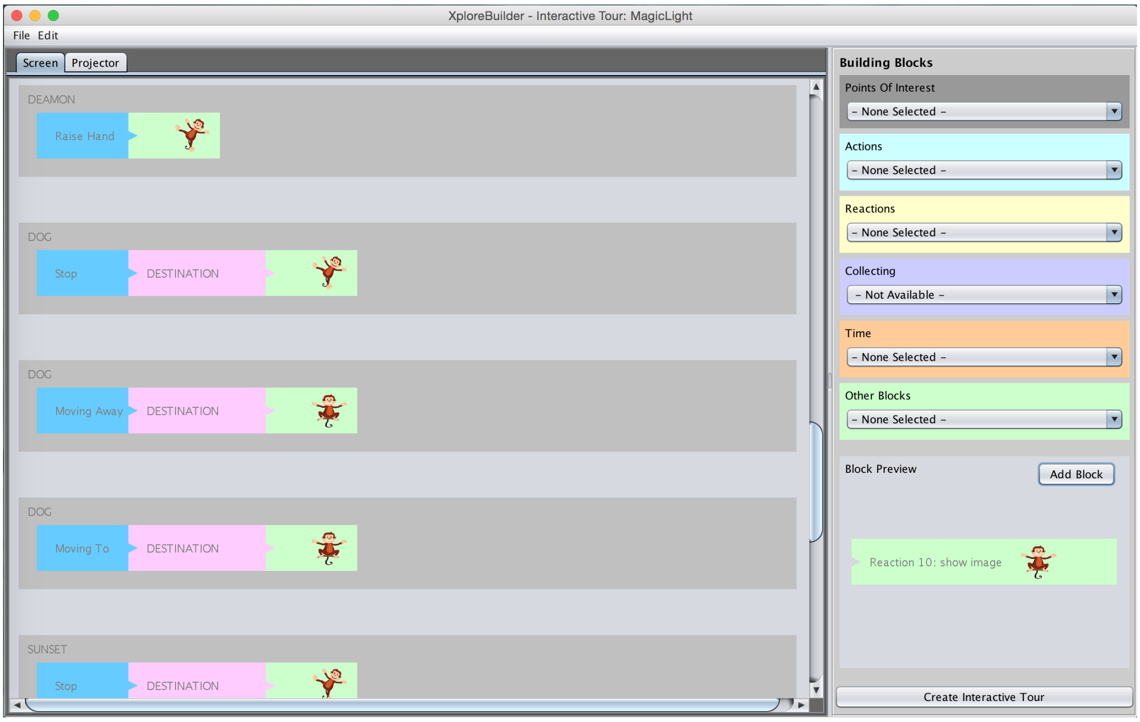
Figure D.1: Design of the UBI, The Guardian Dragon application.

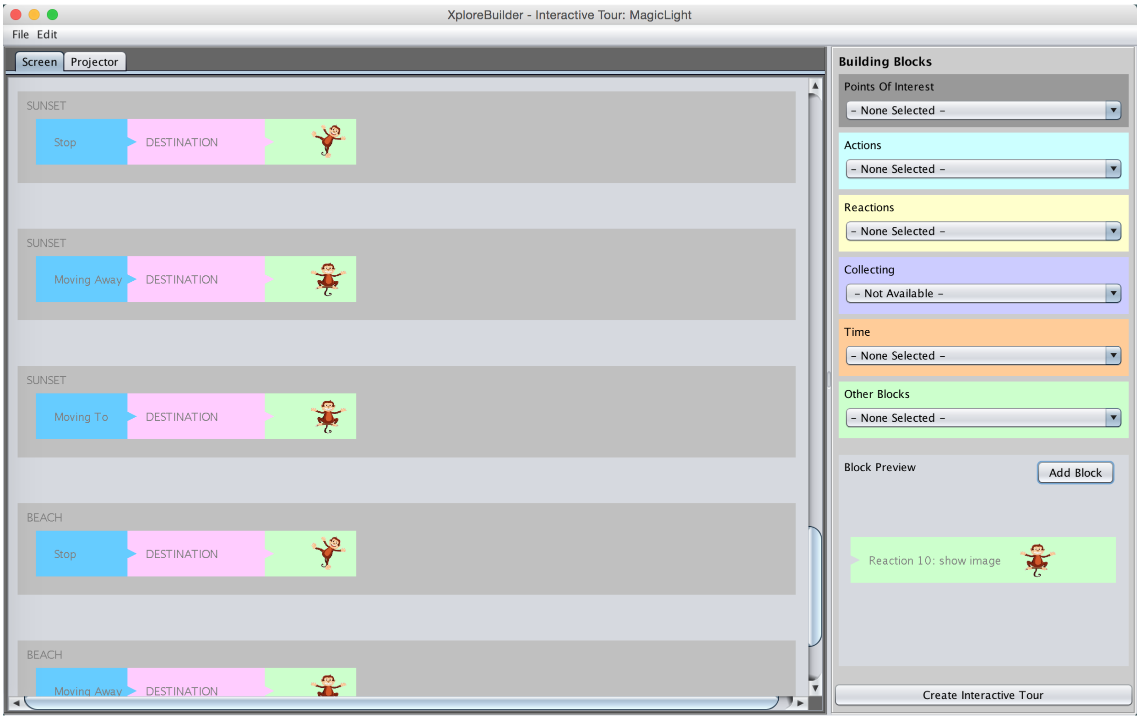
D.2 MagicLight






XploreBuilder - Interactive Tour: MagicLight
File Edit
Screen Projector
CAMERA
Moving To
DEAMON
Out of view
DEAMON
Shake
DURATION 3s
DEAMON
Shake
DURATION 3s
DEAMON
Raise Hand
Building Blocks
Points Of Interest
– None Selected –
Actions
– None Selected –
Reactions
– None Selected –
Collecting
– Not Available –
Time
– None Selected –
Other Blocks
– None Selected –
Block Preview
Add Block
Reaction 10: show image
Create Interactive Tour


XploreBuilder - Interactive Tour: MagicLight
File Edit
Screen Projector
DEAMON
Raise Hand
DOG
Stop
DESTINATION
DOG
Moving Away
DESTINATION
DOG
Moving To
DESTINATION
SUNSET
Stop
DESTINATION
Building Blocks
Points Of Interest
– None Selected –
Actions
– None Selected –
Reactions
– None Selected –
Collecting
– Not Available –
Time
– None Selected –
Other Blocks
– None Selected –
Block Preview
Add Block
Reaction 10: show image
Create Interactive Tour



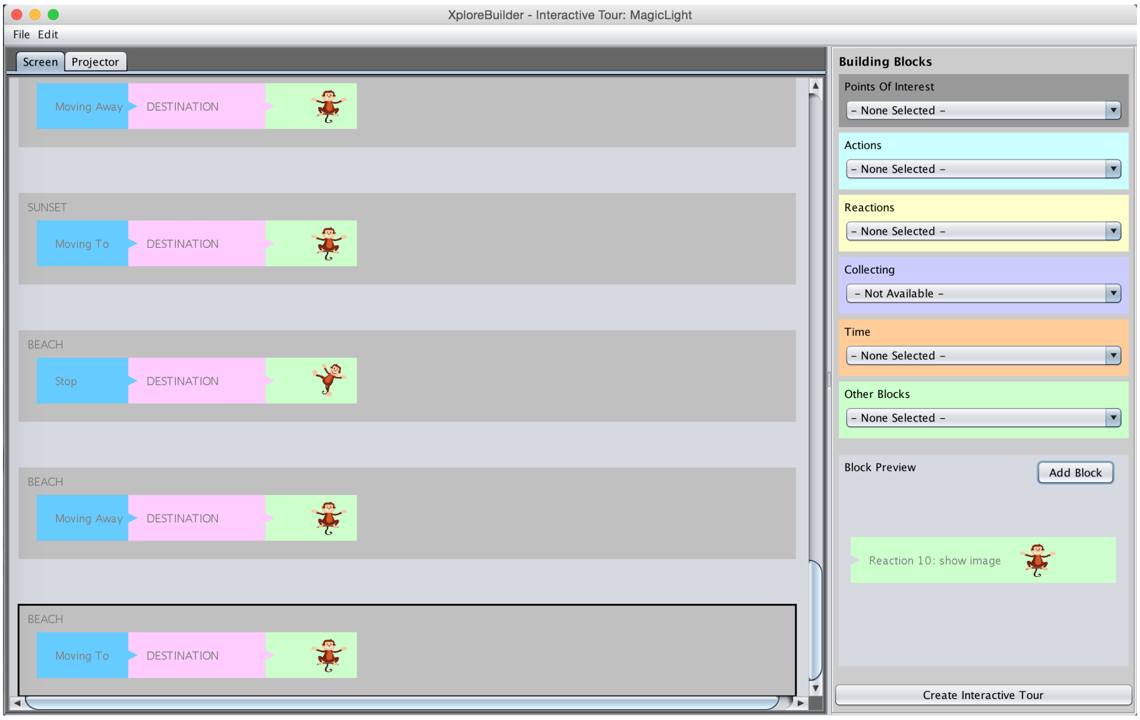
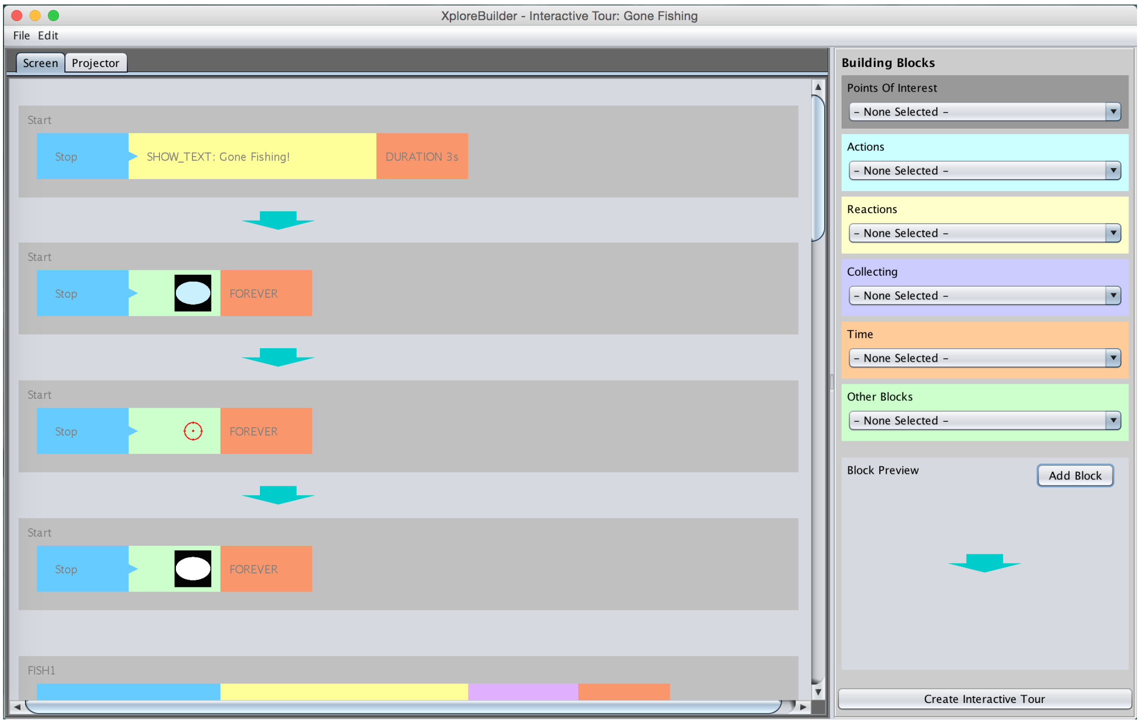
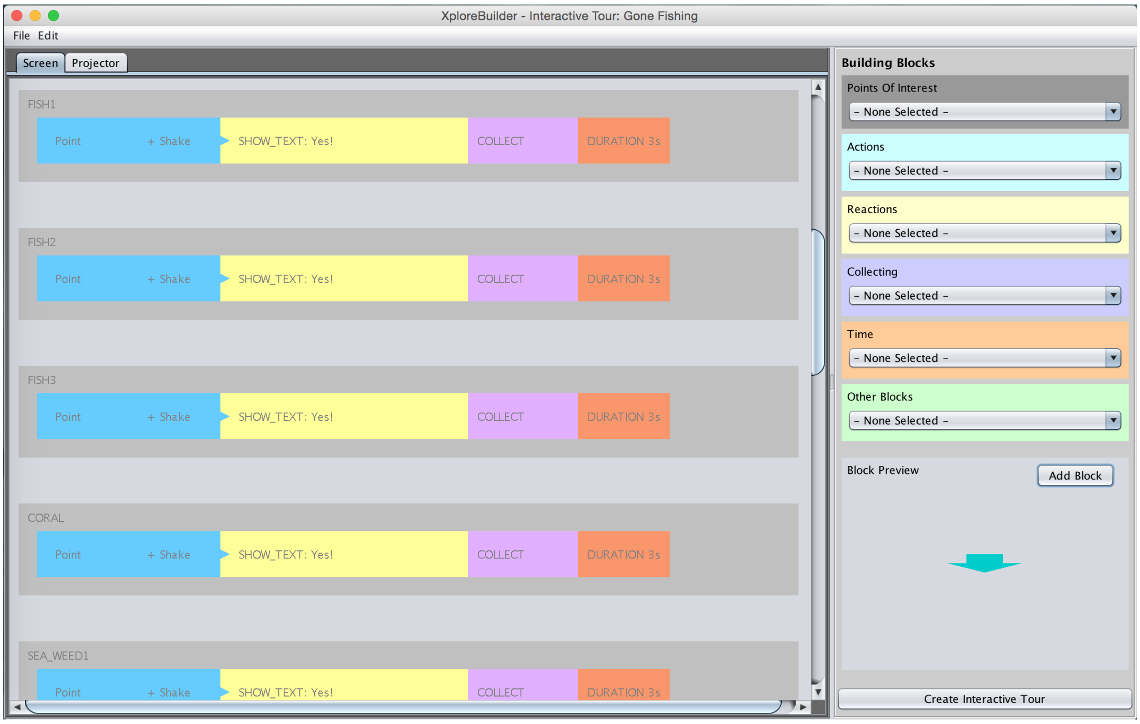
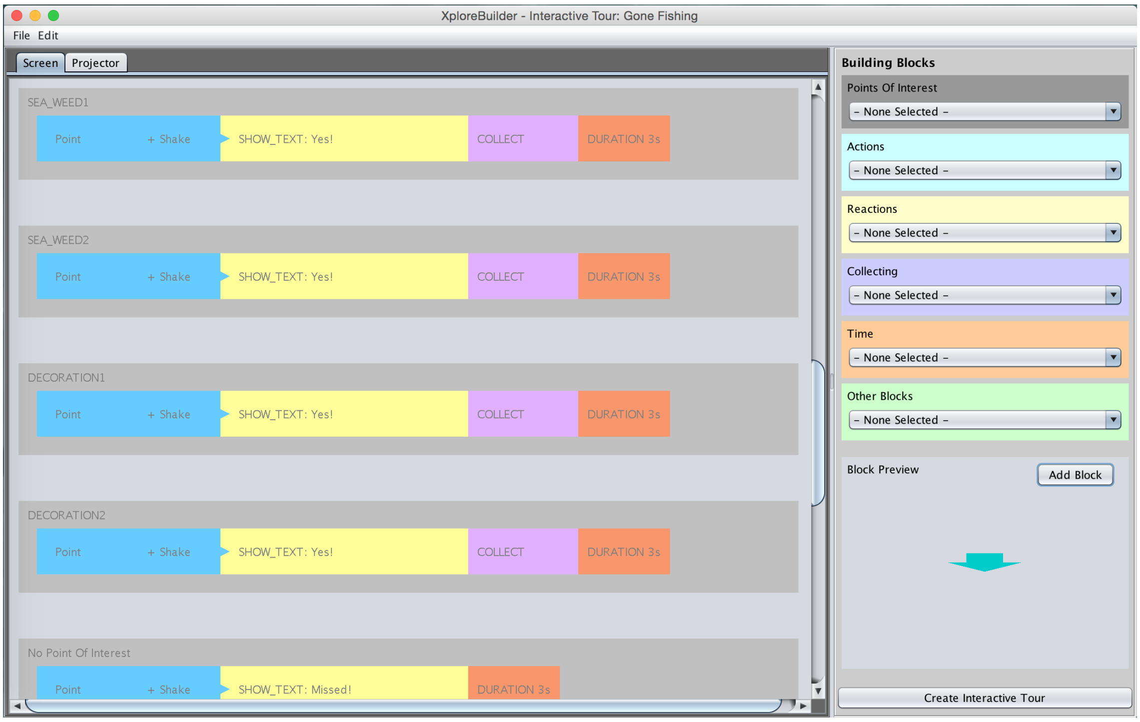


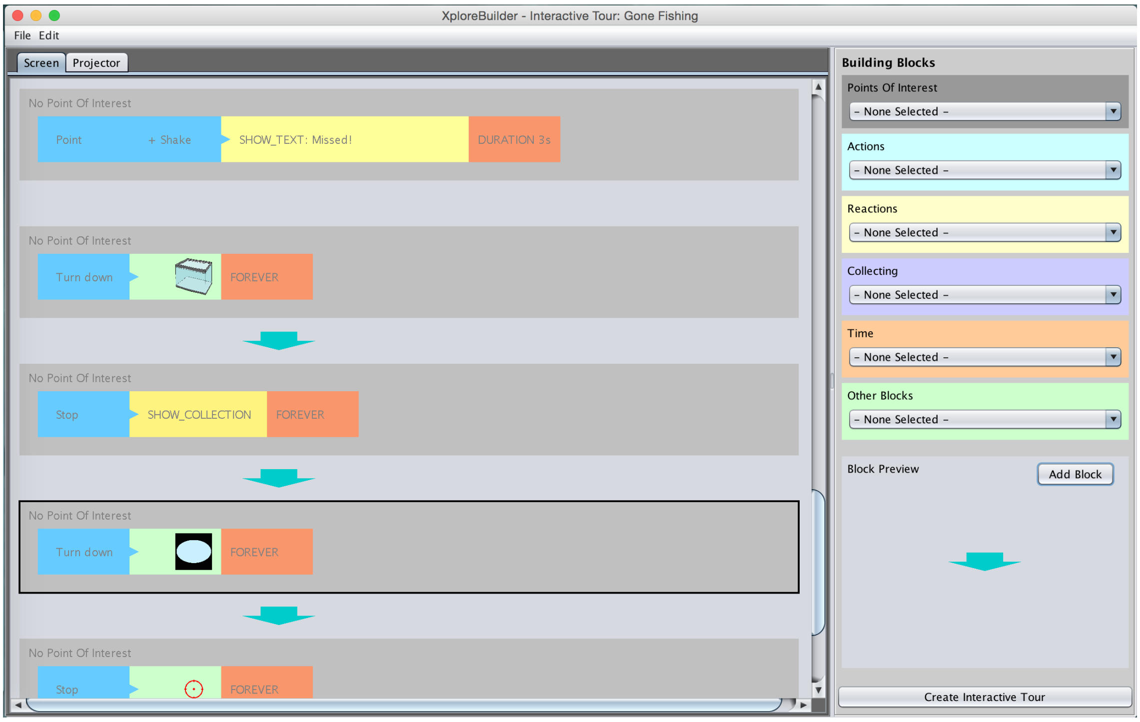
Figure D.2: Design of the MagicLight application.

D.3 Gone Fishing






XploreBuilder - Interactive Tour: Gone Fishing
File Edit
Screen
Projector
SEA_WEED1
Point
+ Shake
SHOW_TEXT: Yes!
COLLECT
DURATION 3s
SEA_WEED2
DECORATION1
DECORATION2
No Point Of Interest
SHOW_TEXT: Missed!
Building Blocks
Points Of Interest
– None Selected –
Actions
Reactions
Collecting
Time
Other Blocks
Block Preview
Add Block
Create Interactive Tour


XploreBuilder - Interactive Tour: Gone Fishing
File Edit
Screen
Projector
No Point Of Interest
Point
+ Shake
SHOW_TEXT: Missed!
DURATION 3s
Turn down
FOREVER
Stop
SHOW_COLLECTION
Building Blocks
Points Of Interest
– None Selected –
Actions
Reactions
Collecting
Time
Other Blocks
Block Preview
Add Block
Create Interactive Tour

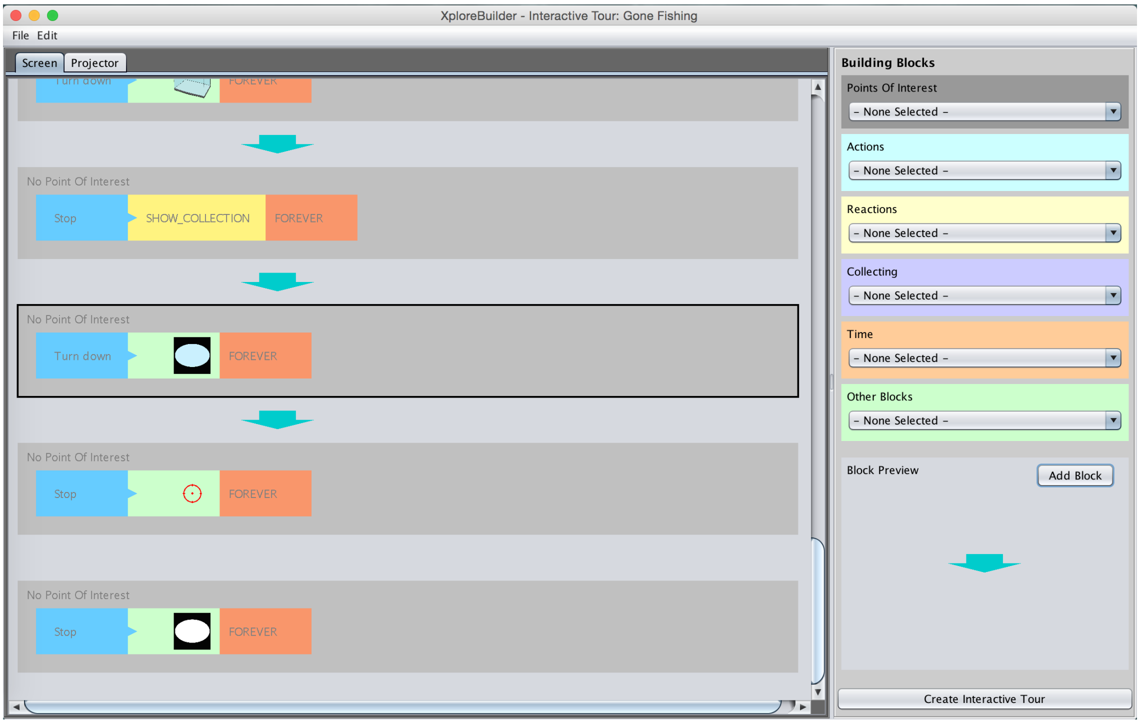
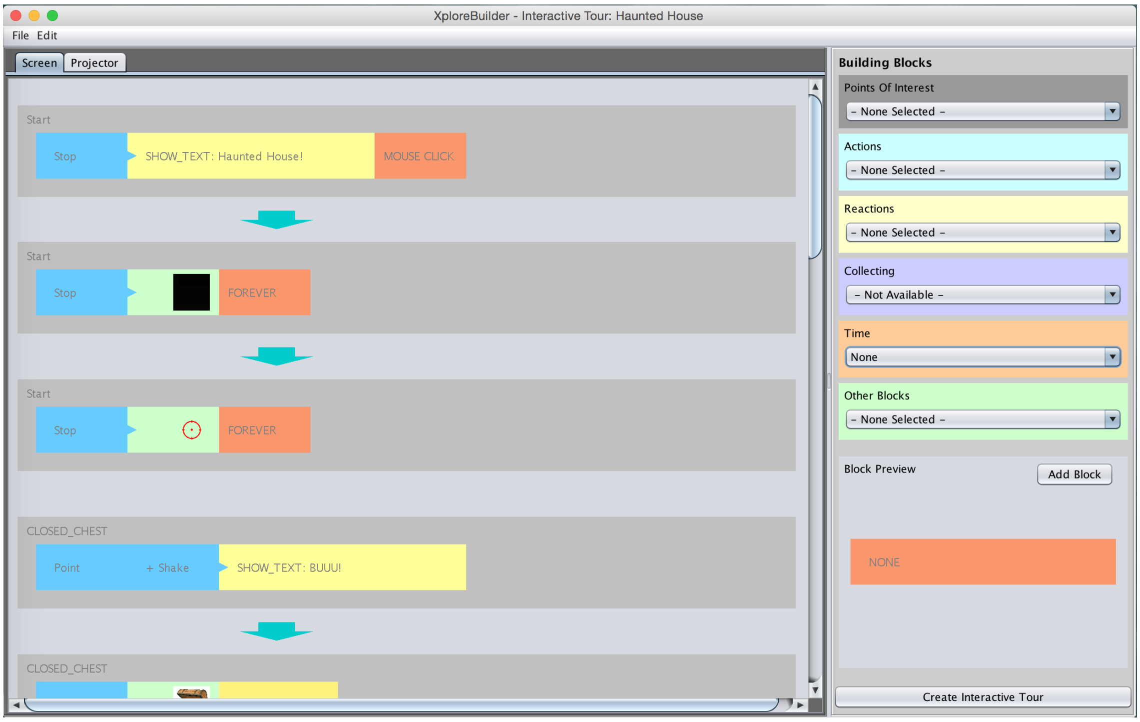


Figure D.3: Design of the Gone Fishing application.

D.4 Haunted House



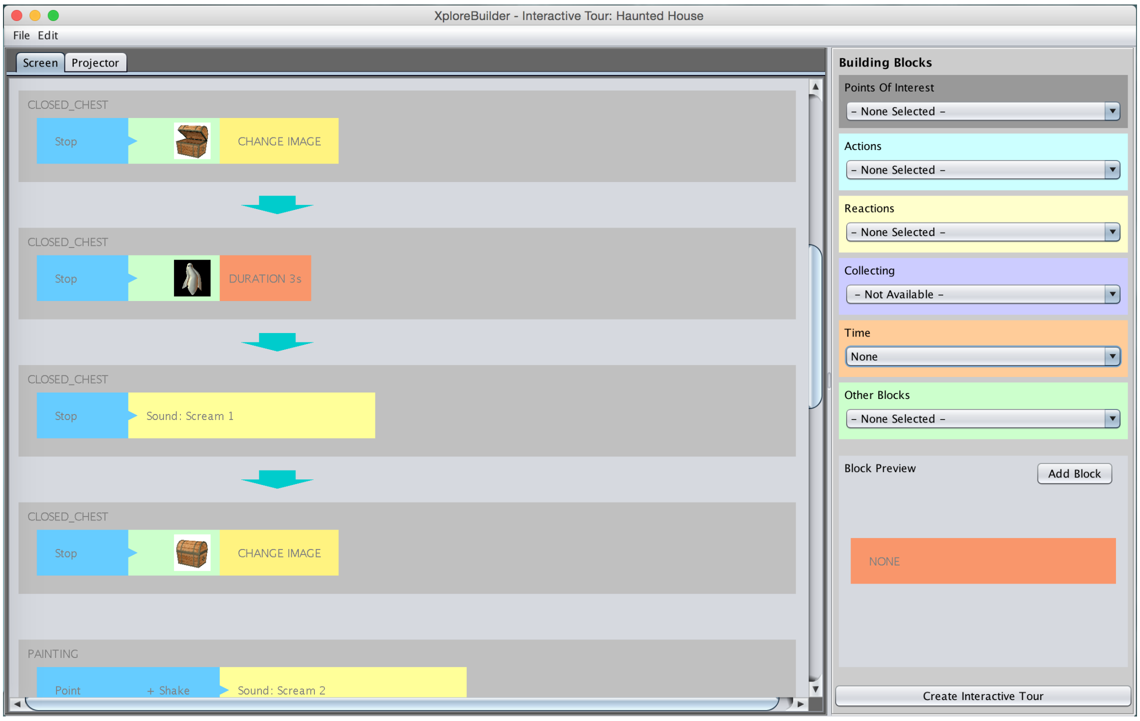


Figure D.4: Design of the Haunted House application.

D.5 Toy Exhibition

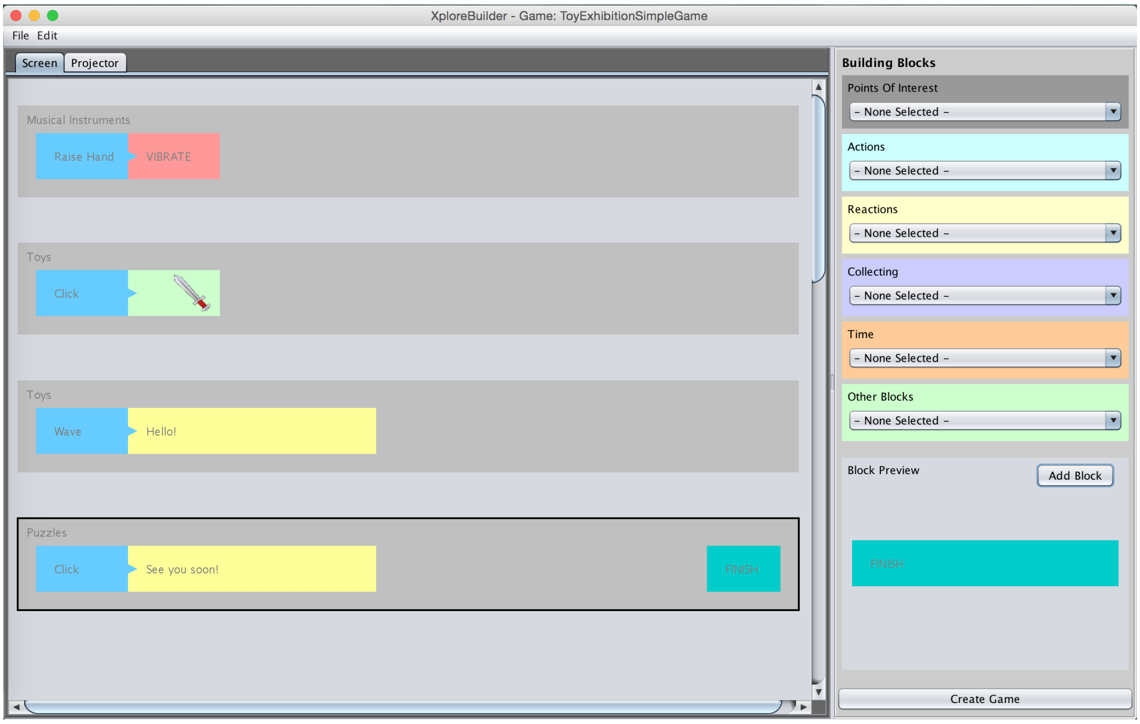


Figure D.5: Design of the Toy Exhibition application.

Augmenting Spaces and Creating Interactive Experiences Using Video Camera Networks

Rossana Santos

2015